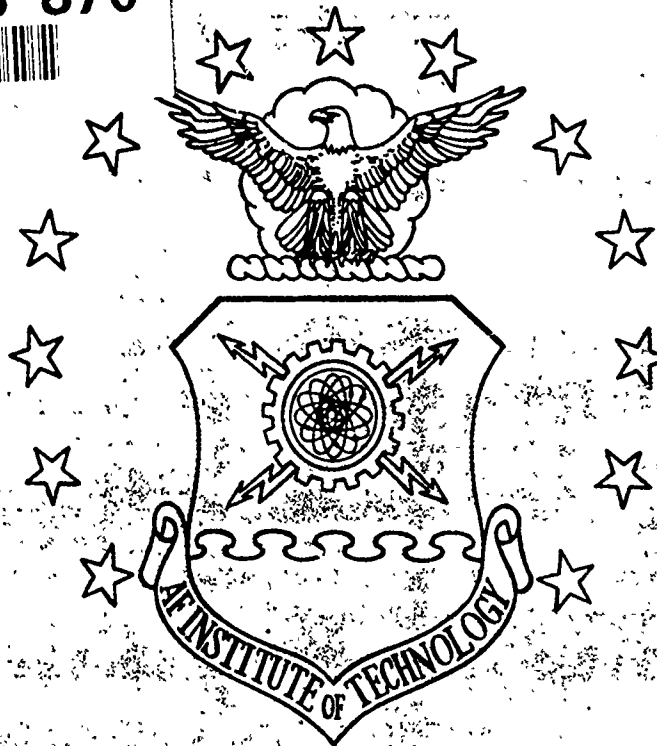


AD-A238 870



(1)



DTIC
FLETC
JUL 2 1971
S D

INVESTIGATION OF ADAPTIVE CONTROLLERS
FOR PUMA TRAJECTORY TRACKING

THESIS

Daniel J. Sims
Captain, USAF

AIR-GL-ENG-91J-05

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

91 7 19 151

AFIT/GE/ENG/91J-05

1

DTIC
ELECTE
JUL 22 1991
S D D

INVESTIGATION OF ADAPTIVE CONTROLLERS
FOR PUMA TRAJECTORY TRACKING

THESIS

Daniel J. Sims
Captain, USAF

AFIT/GE/ENG/91J-05

Approved for public release; distribution unlimited

91-05719



REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 1991	3. REPORT TYPE AND DATES COVERED Thesis	
4. TITLE AND SUBTITLE Investigation of Adaptive Controllers for PUMA Trajectory Tracking.			5. FUNDING NUMBERS	
6. AUTHOR(S) Daniel J. Sims, Capt, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/91J-05	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>Two robust model-based controllers and two decentralized adaptive controllers are experimentally evaluated. Algorithm evaluation is motivated by the need for controllers with good high speed tracking under varying payload conditions. The test case is a PUMA-560 robotic manipulator operating over the standard test suite. The model-based controllers are made robust by the addition of either an auxiliary input term or an adaptive feedforward compensator based on Lyapunov Theory. The model-based auxiliary input controller (MBAIC) adapts the gain matrices used in computing an additional torque to be combined with model-based feedforward and PD feedback torques. The adaptive model-based controllers adapt the assessment of the manipulator parameters used in calculating feedforward torque.</p> <p>The decentralized adaptive controllers are based on either Lyapunov stability or Popov hyperstability. These controllers calculate feedforward, feedback, and auxiliary torques based on trajectory errors and desired trajectory parameters. The gain matrices used to multiply these quantities are adapted. These auxiliary torque components are identical to those used in the MBAICs.</p> <p>Experimental evaluation provides insight into the potential and limitations of each method. The decentralized digital adaptive control algorithms produce an unsatisfactory tracking response. Both model-based control techniques improve the manipulator's tracking response.</p>				
14. SUBJECT TERMS Robot Control, Adaptive Model-Based Control, Model-Based Auxiliary Input Controller, Decentralized Adaptive Control			15. NUMBER OF PAGES 315	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s) Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es) Self-explanatory

Block 10. Sponsoring/Monitoring Agency Report Number (If known)

Block 11. Supplementary Notes Enter information not included elsewhere such as. Prepared in cooperation with... Trans of... To be published in... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

Block 13. Abstract. Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages Enter the total number of pages

Block 16. Price Code. Enter appropriate price code (*NTIS only*).

Blocks 17. - 19. Security Classifications. Self-explanatory Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED) If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited

AFIT/GE/ENG/91J-05

INVESTIGATION OF ADAPTIVE CONTROLLERS
FOR PUMA TRAJECTORY TRACKING

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Daniel J. Sims, BSEE
Captain, USAF

JUNE 1991

Accession For	
NTIS CRASH	LI
DTIC TAB	LI
Unannounced	LI
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved for public release; distribution unlimited



Preface

I would like to express my sincere thanks to my thesis advisor, Capt Michael B. Leahy, for his direction, support, and tolerance during this research effort. I would also like to thank my thesis committee, Dr. Peter Maybeck and Dr. Gary Lamont, for their assistance in the final preparation of this document.

In addition to the my committee members, I'd like to thank Capt Paul Whalen and Mr. Dan Zambon for helping me cure some of my computer illiteracy problems. Special thanks to Captains Bobbie Niblett and Tom Cox for helping me wind up the paperwork to get it out of the door. Finally, I'd like to thank my family for enduring me for the last two years. Thanks, Samantha, for understanding when I couldn't just leave and drop everything for more pleasant endeavors. I especially want to thank my wife, Linda, for her love, understanding, and support through the AFIT ordeal.

Daniel J. Sims

Table of Contents

	Page
Preface	ii
Table of Contents	iii
List of Figures	vii
List of Tables	xxiii
Abstract	xxv
I. Introduction	1-1
1.1 Motivation	1-1
1.2 Problem Statement	1-1
1.3 Research Objective	1-2
1.4 Method of Approach	1-2
1.5 Accomplishments	1-4
1.6 Organization	1-4
II. Literature Review	2-1
2.1 Introduction	2-1
2.2 Adaptive Model-Based Control Algorithm	2-1
2.2.1 Model-Based Control	2-1
2.2.2 Adaptive Model-Based Control	2-3
2.2.3 Experimentation Algorithm	2-5
2.2.4 Friction Models	2-7
2.3 Decentralized Digital Adaptive Control	2-8
2.3.1 Introduction	2-8

	Page
2.3.2 Lyapunov-Based Decentralized Adaptive Control . .	2-8
2.3.3 Discrete-Time Adaptive Control	2-12
2.3.4 Simulation and Experimental Results Discussion . .	2-14
2.4 Summary	2-15
III. Adaptive Model-Based Control	3-1
3.1 Introduction	3-1
3.2 Test Environment	3-1
3.3 Algorithm Implementation	3-2
3.4 Experimental Evaluation	3-4
3.4.1 Introduction	3-4
3.4.2 Sixteen-Parameter Adaptation Testing	3-5
3.4.3 Nineteen-Parameter Adaptation Testing	3-6
3.4.4 Thirteen-Parameter Adaptation Testing	3-6
3.5 Adaptive Model-Based Control Algorithm Test Results . . .	3-7
3.5.1 Introduction	3-7
3.5.2 Single Run Tests With Uninitialized Parameters . .	3-8
3.5.3 Learning Runs With Uninitialized Parameters . . .	3-8
3.5.4 Single Run Tests With Initialized Parameters	3-14
3.5.5 Learning Runs With Initialized Parameters	3-14
3.5.6 Comparison of Initialized and Uninitialized Learning Runs	3-18
3.6 Summary	3-20
IV. Evaluation of Decentralized Digital Control Algorithms	4-1
4.1 Introduction to the Discussion of Tarokh's Algorithm	4-1
4.2 Examination of Tarokh's Algorithm	4-2
4.3 The Tuning Process	4-6
4.3.1 Weighted Error Terms	4-6

	Page
4.3.2 Auxiliary Torque Terms	4-7
4.3.3 Feedforward Torque Terms	4-7
4.3.4 Feedback Torque Terms	4-9
4.4 Examination of Seraji's Algorithm	4-9
4.4.1 Differences in Tarokh's and Seraji's Algorithms . . .	4-12
4.4.2 Experimental Runs Made Using Seraji's Algorithm .	4-13
4.4.3 Evaluation of Tarokh and Seraji Controllers	4-13
4.5 Summary	4-18
V. Model-Based Auxiliary Input Controllers	5-1
5.1 Model-Based Auxiliary Input Controller Implementation . . .	5-1
5.2 Comparison of the Tarokh and Seraji-Based MBAIC	5-2
5.3 Model-Based Auxiliary Input Controller Test Results	5-2
5.4 Comparison of MBAIC and AMBC Controllers	5-3
5.5 Summary	5-3
VI. Conclusions and Recommendations	6-1
6.1 Conclusions	6-1
6.2 Recommendations	6-2
Bibliography	BIB-1
Appendix A. 13-Parameter Uninitialized Learning Runs	A-1
Appendix B. 16-Parameter Uninitialized Learning Runs	B-1
Appendix C. 19-Parameter Uninitialized Learning Runs	C-1
Appendix D. Comparisons of Single Uninitialized Runs	D-1
Appendix E. Comparisons of Uninitialized Runs After Learning	E-1

	Page
Appendix F. 13-Parameter Initialized Learning Runs	F-1
Appendix G. 16-Parameter Initialized Learning Runs	G-1
Appendix H. 19-Parameter Initialized Learning Runs	H-1
Appendix I. Comparisons of Single Initialized Runs	I-1
Appendix J. Comparisons of Initialized Runs After Learning	J-1
Appendix K. Comparisons of 16- and 19-Parameter Learning Runs	K-1
Appendix L. Learned Parameters for 13-Parameter Testing	L-1
Appendix M. Learned Parameters for 16-Parameter Testing	M-1
Appendix N. Learned Parameters for 19-Parameter Testing	N-1
Appendix O. Comparison of Learned Parameters	O-1
Appendix P. Tarokh Algorithm Runs	P-1
Appendix Q. Comparisons of Tarokh and Seraji Control Algorithms . . .	Q-1
Appendix R. Comparisons of MBAIC and SMBC Controllers	R-1
Appendix S. Comparison of MBAIC and AMBC Controllers	S-1
Vita	VITA-1

List of Figures

Figure	Page
3.1. Comparison of Uninitialized Single Run Tests - Trajectory 3	3-9
3.2. Comparison of Uninitialized Single Run Tests - Trajectory 3	3-9
3.3. Comparison of Uninitialized Single Run Tests - Trajectory 3	3-10
3.4. Comparison of Uninitialized Runs After Learning - Trajectory 2	3-11
3.5. Comparison of Uninitialized Runs After Learning - Trajectory 2	3-11
3.6. Comparison of Uninitialized Runs After Learning - Trajectory 2	3-12
3.7. Comparison of Initialized Single Run Testing - Trajectory 4	3-15
3.8. Comparison of Initialized Single Run Testing - Trajectory 4	3-15
3.9. Comparison of Initialized Single Run Testing - Trajectory 4	3-16
3.10. Comparison of Initialized Runs After Learning - Trajectory 4	3-17
3.11. Comparison of Initialized Runs After Learning - Trajectory 4	3-17
3.12. Comparison of Initialized Runs After Learning - Trajectory 4	3-18
3.13. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	3-19
3.14. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	3-19
3.15. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	3-20
4.1. Joint 2 Trajectories with no Joint 3 Movement	4-3
4.2. Joint 2 Trajectories with no Payload	4-3
4.3. Joint 2 Errors with no Joint 3 Movement	4-4
4.4. Joint 2 Errors with no Payload	4-4
4.5. Error Plots During Tuning Tarokh's Algorithm	4-8
4.6. Error Plots During Tuning Tarokh's Algorithm	4-8
4.7. Error Plots During Tuning Tarokh's Algorithm	4-9
4.8. Comparison of Tarokh's and Seraji's Control Algorithms - Trajectory 1	4-14
4.9. Comparison of Tarokh's and Seraji's Control Algorithms - Trajectory 1	4-14

Imagen Laser Printer (iml32)

Owner dsims
Host wa7
printer iml32
Date Wed Apr 24 22:43:29 1991
User dsims
formlength 66

inputbin draft
copies 1
pageduplex off
language ultrascript
jobheader on

System Version TURBO UltraScript 5.0T IP/II, Serial #89:10:51
Page images processed: 8
Pages printed: 8

Paper size (width, height):
2560, 3328
Document length:
70809 bytes

%%[Error: syntaxerror; OffendingCommand: --nostringval--]%%

AFIT/GE/ENG/91J-05

INVESTIGATION OF ADAPTIVE CONTROLLERS
FOR PUMA TRAJECTORY TRACKING

THESIS

Daniel J. Sims
Captain, USAF

AFIT/GE/ENG/91J-05

Approved for public release; distribution unlimited

AFIT/GE/ENG/91J-05

INVESTIGATION OF ADAPTIVE CONTROLLERS
FOR PUMA TRAJECTORY TRACKING

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Daniel J. Sims, BSEE
Captain, USAF

JUNE 1991

Approved for public release; distribution unlimited

Preface

I would like to express my sincere thanks to my thesis advisor, Capt Michael B. Leahy, for his direction, support, and tolerance during this research effort. I would also like to thank my thesis committee, Dr. Peter Maybeck and Dr. Gary Lamont, for their assistance in the final preparation of this document.

In addition to the my committee members, I'd like to thank Capt Paul Whalen and Mr. Dan Zambon for helping me cure some of my computer illiteracy problems. Special thanks to Captains Bobbie Niblett and Tom Cox for helping me wind up the paperwork to get it out of the door. Finally, I'd like to thank my family for enduring me for the last two years. Thanks, Samantha, for understanding when I couldn't just leave and drop everything for more pleasant endeavors. I especially want to thank my wife, Linda, for her love, understanding, and support through the AFIT ordeal.

Daniel J. Sims

Table of Contents

	Page
Preface	ii
Table of Contents	iii
List of Figures	vii
List of Tables	xxiii
Abstract	xxv
I. Introduction	1-1
1.1 Motivation	1-1
1.2 Problem Statement	1-1
1.3 Research Objective	1-2
1.4 Method of Approach	1-2
1.5 Accomplishments	1-4
1.6 Organization	1-4
II. Literature Review	2-1
2.1 Introduction	2-1
2.2 Adaptive Model-Based Control Algorithm	2-1
2.2.1 Model-Based Control	2-1
2.2.2 Adaptive Model-Based Control	2-3
2.2.3 Experimentation Algorithm	2-5
2.2.4 Friction Models	2-7
2.3 Decentralized Digital Adaptive Control	2-8
2.3.1 Introduction	2-8

	Page
2.3.2 Lyapunov-Based Decentralized Adaptive Control . .	2-8
2.3.3 Discrete-Time Adaptive Control	2-12
2.3.4 Simulation and Experimental Results Discussion . .	2-14
2.4 Summary	2-15
III. Adaptive Model-Based Control	3-1
3.1 Introduction	3-1
3.2 Test Environment	3-1
3.3 Algorithm Implementation	3-2
3.4 Experimental Evaluation	3-4
3.4.1 Introduction	3-4
3.4.2 Sixteen-Parameter Adaptation Testing	3-5
3.4.3 Nineteen-Parameter Adaptation Testing	3-6
3.4.4 Thirteen-Parameter Adaptation Testing	3-6
3.5 Adaptive Model-Based Control Algorithm Test Results . . .	3-7
3.5.1 Introduction	3-7
3.5.2 Single Run Tests With Uninitialized Parameters . .	3-8
3.5.3 Learning Runs With Uninitialized Parameters . . .	3-8
3.5.4 Single Run Tests With Initialized Parameters	3-14
3.5.5 Learning Runs With Initialized Parameters	3-14
3.5.6 Comparison of Initialized and Uninitialized Learning Runs	3-18
3.6 Summary	3-20
IV. Evaluation of Decentralized Digital Control Algorithms	4-1
4.1 Introduction to the Discussion of Tarokh's Algorithm	4-1
4.2 Examination of Tarokh's Algorithm	4-2
4.3 The Tuning Process	4-6
4.3.1 Weighted Error Terms	4-6

	Page
4.3.2 Auxiliary Torque Terms	4-7
4.3.3 Feedforward Torque Terms	4-7
4.3.4 Feedback Torque Terms	4-9
4.4 Examination of Seraji's Algorithm	4-9
4.4.1 Differences in Tarokh's and Seraji's Algorithms . . .	4-12
4.4.2 Experimental Runs Made Using Seraji's Algorithm .	4-13
4.4.3 Evaluation of Tarokh and Seraji Controllers	4-13
4.5 Summary	4-18
V. Model-Based Auxiliary Input Controllers	5-1
5.1 Model-Based Auxiliary Input Controller Implementation . . .	5-1
5.2 Comparison of the Tarokh and Seraji-Based MBAIC	5-2
5.3 Model-Based Auxiliary Input Controller Test Results	5-2
5.4 Comparison of MBAIC and AMBC Controllers	5-3
5.5 Summary	5-3
VI. Conclusions and Recommendations	6-1
6.1 Conclusions	6-1
6.2 Recommendations	6-2
Bibliography	BIB-1
Appendix A. 13-Parameter Uninitialized Learning Runs	A-1
Appendix B. 16-Parameter Uninitialized Learning Runs	B-1
Appendix C. 19-Parameter Uninitialized Learning Runs	C-1
Appendix D. Comparisons of Single Uninitialized Runs	D-1
Appendix E. Comparisons of Uninitialized Runs After Learning	E-1

	Page
Appendix F. 13-Parameter Initialized Learning Runs	F-1
Appendix G. 16-Parameter Initialized Learning Runs	G-1
Appendix H. 19-Parameter Initialized Learning Runs	H-1
Appendix I. Comparisons of Single Initialized Runs	I-1
Appendix J. Comparisons of Initialized Runs After Learning	J-1
Appendix K. Comparisons of 16- and 19-Parameter Learning Runs	K-1
Appendix L. Learned Parameters for 13-Parameter Testing	L-1
Appendix M. Learned Parameters for 16-Parameter Testing	M-1
Appendix N. Learned Parameters for 19-Parameter Testing	N-1
Appendix O. Comparison of Learned Parameters	O-1
Appendix P. Tarokh Algorithm Runs	P-1
Appendix Q. Comparisons of Tarokh and Seraji Control Algorithms	Q-1
Appendix R. Comparisons of MBAIC and SMBC Controllers	R-1
Appendix S. Comparison of MBAIC and AMBC Controllers	S-1
Vita	VITA-1

List of Figures

Figure	Page
3.1. Comparison of Uninitialized Single Run Tests - Trajectory 3	3-9
3.2. Comparison of Uninitialized Single Run Tests - Trajectory 3	3-9
3.3. Comparison of Uninitialized Single Run Tests - Trajectory 3	3-10
3.4. Comparison of Uninitialized Runs After Learning - Trajectory 2	3-11
3.5. Comparison of Uninitialized Runs After Learning - Trajectory 2	3-11
3.6. Comparison of Uninitialized Runs After Learning - Trajectory 2	3-12
3.7. Comparison of Initialized Single Run Testing - Trajectory 4	3-15
3.8. Comparison of Initialized Single Run Testing - Trajectory 4	3-15
3.9. Comparison of Initialized Single Run Testing - Trajectory 4	3-16
3.10. Comparison of Initialized Runs After Learning - Trajectory 4	3-17
3.11. Comparison of Initialized Runs After Learning - Trajectory 4	3-17
3.12. Comparison of Initialized Runs After Learning - Trajectory 4	3-18
3.13. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	3-19
3.14. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	3-19
3.15. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	3-20
4.1. Joint 2 Trajectories with no Joint 3 Movement	4-3
4.2. Joint 2 Trajectories with no Payload	4-3
4.3. Joint 2 Errors with no Joint 3 Movement	4-4
4.4. Joint 2 Errors with no Payload	4-4
4.5. Error Plots During Tuning Tarokh's Algorithm	4-8
4.6. Error Plots During Tuning Tarokh's Algorithm	4-8
4.7. Error Plots During Tuning Tarokh's Algorithm	4-9
4.8. Comparison of Tarokh's and Seraji's Control Algorithms - Trajectory 1	4-14
4.9. Comparison of Tarokh's and Seraji's Control Algorithms - Trajectory 1	4-14

Figure	Page
4.10. Comparison of Tarokh's and Seraji's Control Algorithms - Trajectory 1	4-15
4.11. Effects of Tarokh Discrete Adaptive Controller's Trajectory 0 Parameters on Trajectories 2 and 3	4-16
4.12. Effects of Tarokh Discrete Adaptive Controller's Trajectory 0 Parameters on Trajectories 2 and 3	4-16
4.13. The Effects of Trajectory 0 Tuning Parameters on Trajectories 2 and 3	4-17
4.14. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5	4-19
4.15. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5	4-19
4.16. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5	4-20
4.17. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5	4-20
4.18. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6	4-21
4.19. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6	4-21
4.20. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec	4-22
4.21. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec	4-22
4.22. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec	4-23
5.1. Comparison of MBAIC and SMBC Controllers - Trajectory 0	5-4
5.2. Comparison of MBAIC and SMBC Controllers - Trajectory 0	5-4
5.3. Comparison of MBAIC and SMBC Controllers - Trajectory 0	5-5
A.1. 13-Parameter Uninitialized Adaptation Runs - Trajectory 2	A-2
A.2. 13-Parameter Uninitialized Adaptation Runs - Trajectory 2	A-2
A.3. 13-Parameter Uninitialized Adaptation Runs - Trajectory 2	A-3
A.4. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3	A-3
A.5. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3	A-4
A.6. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3	A-4
A.7. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	A-5
A.8. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	A-5
A.9. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	A-6

Figure	Page
A.10.13-Parameter Uninitialized Adaptation Runs - Trajectory 4	A-6
A.11.13-Parameter Uninitialized Adaptation Runs - Trajectory 4	A-7
A.12.13-Parameter Uninitialized Adaptation Runs - Trajectory 4	A-7
A.13.13-Parameter Uninitialized Adaptation Runs - Trajectory 5	A-8
A.14.13-Parameter Uninitialized Adaptation Runs - Trajectory 5	A-8
A.15.13-Parameter Uninitialized Adaptation Runs - Trajectory 5	A-9
A.16.13-Parameter Uninitialized Adaptation Runs - Trajectory 1	A-9
A.17.13-Parameter Uninitialized Adaptation Runs - Trajectory 1	A-10
A.18.13-Parameter Uninitialized Adaptation Runs - Trajectory 1	A-10
A.19.13-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	A-11
A.20.13-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	A-11
A.21.13-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	A-12
A.22.13-Parameter Uninitialized Adaptation Runs - Trajectory 6	A-12
A.23.13-Parameter Uninitialized Adaptation Runs - Trajectory 6	A-13
A.24.13-Parameter Uninitialized Adaptation Runs - Trajectory 6	A-13
B.1. 16-Parameter Uninitialized Adaptation Runs - Trajectory 2	B-2
B.2. 16-Parameter Uninitialized Adaptation Runs - Trajectory 2	B-2
B.3. 16-Parameter Uninitialized Adaptation Runs - Trajectory 2	B-3
B.4. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3	B-3
B.5. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3	B-4
B.6. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3	B-4
B.7. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	B-5
B.8. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	B-5
B.9. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	B-6
B.10.16-Parameter Uninitialized Adaptation Runs - Trajectory 4	B-6
B.11.16-Parameter Uninitialized Adaptation Runs - Trajectory 4	B-7
B.12.16-Parameter Uninitialized Adaptation Runs - Trajectory 4	B-7

Figure	Page
B.13.16-Parameter Uninitialized Adaptation Runs - Trajectory 5	B-8
B.14.16-Parameter Uninitialized Adaptation Runs - Trajectory 5	B-8
B.15.16-Parameter Uninitialized Adaptation Runs - Trajectory 5	B-9
B.16.16-Parameter Uninitialized Adaptation Runs - Trajectory 1	B-9
B.17.16-Parameter Uninitialized Adaptation Runs - Trajectory 1	B-10
B.18.16-Parameter Uninitialized Adaptation Runs - Trajectory 1	B-10
B.19.16-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	B-11
B.20.16-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	B-11
B.21.16-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	B-12
B.22.16-Parameter Uninitialized Adaptation Runs - Trajectory 6	B-12
B.23.16-Parameter Uninitialized Adaptation Runs - Trajectory 6	B-13
B.24.16-Parameter Uninitialized Adaptation Runs - Trajectory 6	B-13
C.1. 19-Parameter Uninitialized Adaptation Runs - Trajectory 2	C-2
C.2. 19-Parameter Uninitialized Adaptation Runs - Trajectory 2	C-2
C.3. 19-Parameter Uninitialized Adaptation Runs - Trajectory 2	C-3
C.4. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3	C-3
C.5. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3	C-4
C.6. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3	C-4
C.7. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	C-5
C.8. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	C-5
C.9. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload	C-6
C.10.19-Parameter Uninitialized Adaptation Runs - Trajectory 4	C-6
C.11.19-Parameter Uninitialized Adaptation Runs - Trajectory 4	C-7
C.12.19-Parameter Uninitialized Adaptation Runs - Trajectory 4	C-7
C.13.19-Parameter Uninitialized Adaptation Runs - Trajectory 5	C-8
C.14.19-Parameter Uninitialized Adaptation Runs - Trajectory 5	C-8
C.15.19-Parameter Uninitialized Adaptation Runs - Trajectory 5	C-9

Figure	Page
C.16.19-Parameter Uninitialized Adaptation Runs - Trajectory 1	C-9
C.17.19-Parameter Uninitialized Adaptation Runs - Trajectory 1	C-10
C.18.19-Parameter Uninitialized Adaptation Runs - Trajectory 1	C-10
C.19.19-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	C-11
C.20.19-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	C-11
C.21.19-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload	C-12
C.22.19-Parameter Uninitialized Adaptation Runs - Trajectory 6	C-12
C.23.19-Parameter Uninitialized Adaptation Runs - Trajectory 6	C-13
C.24.19-Parameter Uninitialized Adaptation Runs - Trajectory 6	C-13
D.1. Comparison of Single Uninitialized Runs - Trajectory 2	D-2
D.2. Comparison of Single Uninitialized Runs - Trajectory 2	D-2
D.3. Comparison of Single Uninitialized Runs - Trajectory 2	D-3
D.4. Comparison of Single Uninitialized Runs - Trajectory 3	D-3
D.5. Comparison of Single Uninitialized Runs - Trajectory 3	D-4
D.6. Comparison of Single Uninitialized Runs - Trajectory 3	D-4
D.7. Comparison of Single Uninitialized Runs - Trajectory 3 w/ Payload . .	D-5
D.8. Comparison of Single Uninitialized Runs - Trajectory 3 w/ Payload . .	D-5
D.9. Comparison of Single Uninitialized Runs - Trajectory 3 w/ Payload . .	D-6
D.10.Comparison of Single Uninitialized Runs - Trajectory 4	D-6
D.11.Comparison of Single Uninitialized Runs - Trajectory 4	D-7
D.12.Comparison of Single Uninitialized Runs - Trajectory 4	D-7
D.13.Comparison of Single Uninitialized Runs - Trajectory 5	D-8
D.14.Comparison of Single Uninitialized Runs - Trajectory 5	D-8
D.15.Comparison of Single Uninitialized Runs - Trajectory 5	D-9
D.16.Comparison of Single Uninitialized Runs - Trajectory 1	D-9
D.17.Comparison of Single Uninitialized Runs - Trajectory 1	D-10
D.18.Comparison of Single Uninitialized Runs - Trajectory 1	D-10

Figure	Page
D.19.Comparison of Single Uninitialized Runs - Trajectory 1 w/ Payload . .	D-11
D.20.Comparison of Single Uninitialized Runs - Trajectory 1 w/ Payload . .	D-11
D.21.Comparison of Single Uninitialized Runs - Trajectory 1 w/ Payload . .	D-12
D.22.Comparison of Single Uninitialized Runs - Trajectory 6	D-12
D.23.Comparison of Single Uninitialized Runs - Trajectory 6	D-13
D.24.Comparison of Single Uninitialized Runs - Trajectory 6	D-13
E.1. Comparison of Uninitialized Runs After Learning - Trajectory 2	E-2
E.2. Comparison of Uninitialized Runs After Learning - Trajectory 2	E-2
E.3. Comparison of Uninitialized Runs After Learning - Trajectory 2	E-3
E.4. Comparison of Uninitialized Runs After Learning - Trajectory 3	E-3
E.5. Comparison of Uninitialized Runs After Learning - Trajectory 3	E-4
E.6. Comparison of Uninitialized Runs After Learning - Trajectory 3	E-4
E.7. Comparison of Uninitialized Runs After Learning - Trajectory 3 w/ Payload	E-5
E.8. Comparison of Uninitialized Runs After Learning - Trajectory 3 w/ Payload	E-5
E.9. Comparison of Uninitialized Runs After Learning - Trajectory 3 w/ Payload	E-6
E.10.Comparison of Uninitialized Runs After Learning - Trajectory 4	E-6
E.11.Comparison of Uninitialized Runs After Learning - Trajectory 4	E-7
E.12.Comparison of Uninitialized Runs After Learning - Trajectory 4	E-7
E.13.Comparison of Uninitialized Runs After Learning - Trajectory 5	E-8
E.14.Comparison of Uninitialized Runs After Learning - Trajectory 5	E-8
E.15.Comparison of Uninitialized Runs After Learning - Trajectory 5	E-9
E.16.Comparison of Uninitialized Runs After Learning - Trajectory 1	E-9
E.17.Comparison of Uninitialized Runs After Learning - Trajectory 1	E-10
E.18.Comparison of Uninitialized Runs After Learning - Trajectory 1	E-10
E.19.Comparison of Uninitialized Runs After Learning - Trajectory 1 w/ Payload	E-11
E.20.Comparison of Uninitialized Runs After Learning - Trajectory 1 w/ Payload	E-11
E.21.Comparison of Uninitialized Runs After Learning - Trajectory 1 w/ Payload	E-12

Figure	Page
E.22.Comparison of Uninitialized Runs After Learning - Trajectory 6	E-12
E.23.Comparison of Uninitialized Runs After Learning - Trajectory 6	E-13
E.24.Comparison of Uninitialized Runs After Learning - Trajectory 6	E-13
F.1. 13-Parameter Initialized Adaptation Runs - Trajectory 2	F-2
F.2. 13-Parameter Initialized Adaptation Runs - Trajectory 2	F-2
F.3. 13-Parameter Initialized Adaptation Runs - Trajectory 2	F-3
F.4. 13-Parameter Initialized Adaptation Runs - Trajectory 3	F-3
F.5. 13-Parameter Initialized Adaptation Runs - Trajectory 3	F-4
F.6. 13-Parameter Initialized Adaptation Runs - Trajectory 3	F-4
F.7. 13-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	F-5
F.8. 13-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	F-5
F.9. 13-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	F-6
F.10.13-Parameter Initialized Adaptation Runs - Trajectory 4	F-6
F.11.13-Parameter Initialized Adaptation Runs - Trajectory 4	F-7
F.12.13-Parameter Initialized Adaptation Runs - Trajectory 4	F-7
F.13.13-Parameter Initialized Adaptation Runs - Trajectory 5	F-8
F.14.13-Parameter Initialized Adaptation Runs - Trajectory 5	F-8
F.15.13-Parameter Initialized Adaptation Runs - Trajectory 5	F-9
F.16.13-Parameter Initialized Adaptation Runs - Trajectory 1	F-9
F.17.13-Parameter Initialized Adaptation Runs - Trajectory 1	F-10
F.18.13-Parameter Initialized Adaptation Runs - Trajectory 1	F-10
F.19.13-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	F-11
F.20.13-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	F-11
F.21.13-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	F-12
F.22.13-Parameter Initialized Adaptation Runs - Trajectory 6	F-12
F.23.13-Parameter Initialized Adaptation Runs - Trajectory 6	F-13
F.24.13-Parameter Initialized Adaptation Runs - Trajectory 6	F-13

Figure	Page
G.1. 16-Parameter Initialized Adaptation Runs - Trajectory 2	G-2
G.2. 16-Parameter Initialized Adaptation Runs - Trajectory 2	G-2
G.3. 16-Parameter Initialized Adaptation Runs - Trajectory 2	G-3
G.4. 16-Parameter Initialized Adaptation Runs - Trajectory 3	G-3
G.5. 16-Parameter Initialized Adaptation Runs - Trajectory 3	G-4
G.6. 16-Parameter Initialized Adaptation Runs - Trajectory 3	G-4
G.7. 16-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	G-5
G.8. 16-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	G-5
G.9. 16-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	G-6
G.10. 16-Parameter Initialized Adaptation Runs - Trajectory 4	G-6
G.11. 16-Parameter Initialized Adaptation Runs - Trajectory 4	G-7
G.12. 16-Parameter Initialized Adaptation Runs - Trajectory 4	G-7
G.13. 16-Parameter Initialized Adaptation Runs - Trajectory 5	G-8
G.14. 16-Parameter Initialized Adaptation Runs - Trajectory 5	G-8
G.15. 16-Parameter Initialized Adaptation Runs - Trajectory 5	G-9
G.16. 16-Parameter Initialized Adaptation Runs - Trajectory 1	G-9
G.17. 16-Parameter Initialized Adaptation Runs - Trajectory 1	G-10
G.18. 16-Parameter Initialized Adaptation Runs - Trajectory 1	G-10
G.19. 16-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	G-11
G.20. 16-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	G-11
G.21. 16-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	G-12
G.22. 16-Parameter Initialized Adaptation Runs - Trajectory 6	G-12
G.23. 16-Parameter Initialized Adaptation Runs - Trajectory 6	G-13
G.24. 16-Parameter Initialized Adaptation Runs - Trajectory 6	G-13
H.1. 19-Parameter Initialized Adaptation Runs - Trajectory 2	H-2
H.2. 19-Parameter Initialized Adaptation Runs - Trajectory 2	H-2
H.3. 19-Parameter Initialized Adaptation Runs - Trajectory 2	H-3

Figure	Page
H.4. 19-Parameter Initialized Adaptation Runs - Trajectory 3	H-3
H.5. 19-Parameter Initialized Adaptation Runs - Trajectory 3	H-4
H.6. 19-Parameter Initialized Adaptation Runs - Trajectory 3	H-4
H.7. 19-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	H-5
H.8. 19-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	H-5
H.9. 19-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload .	H-6
H.10. 19-Parameter Initialized Adaptation Runs - Trajectory 4	H-6
H.11. 19-Parameter Initialized Adaptation Runs - Trajectory 4	H-7
H.12. 19-Parameter Initialized Adaptation Runs - Trajectory 4	H-7
H.13. 19-Parameter Initialized Adaptation Runs - Trajectory 5	H-8
H.14. 19-Parameter Initialized Adaptation Runs - Trajectory 5	H-8
H.15. 19-Parameter Initialized Adaptation Runs - Trajectory 5	H-9
H.16. 19-Parameter Initialized Adaptation Runs - Trajectory 1	H-9
H.17. 19-Parameter Initialized Adaptation Runs - Trajectory 1	H-10
H.18. 19-Parameter Initialized Adaptation Runs - Trajectory 1	H-10
H.19. 19-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	H-11
H.20. 19-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	H-11
H.21. 19-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload .	H-12
H.22. 19-Parameter Initialized Adaptation Runs - Trajectory 6	H-12
H.23. 19-Parameter Initialized Adaptation Runs - Trajectory 6	H-13
H.24. 19-Parameter Initialized Adaptation Runs - Trajectory 6	H-13
I.1. Comparison of Single Initialized Runs - Trajectory 2	I-2
I.2. Comparison of Single Initialized Runs - Trajectory 2	I-2
I.3. Comparison of Single Initialized Runs - Trajectory 2	I-3
I.4. Comparison of Single Initialized Runs - Trajectory 3	I-3
I.5. Comparison of Single Initialized Runs - Trajectory 3	I-4
I.6. Comparison of Single Initialized Runs - Trajectory 3	I-4

Figure	Page
I.7. Comparison of Single Initialized Runs - Trajectory 3 w/ Payload	I-5
I.8. Comparison of Single Initialized Runs - Trajectory 3 w/ Payload	I-5
I.9. Comparison of Single Initialized Runs - Trajectory 3 w/ Payload	I-6
I.10. Comparison of Single Initialized Runs - Trajectory 4	I-6
I.11. Comparison of Single Initialized Runs - Trajectory 4	I-7
I.12. Comparison of Single Initialized Runs - Trajectory 4	I-7
I.13. Comparison of Single Initialized Runs - Trajectory 5	I-8
I.14. Comparison of Single Initialized Runs - Trajectory 5	I-8
I.15. Comparison of Single Initialized Runs - Trajectory 5	I-9
I.16. Comparison of Single Initialized Runs - Trajectory 1	I-9
I.17. Comparison of Single Initialized Runs - Trajectory 1	I-10
I.18. Comparison of Single Initialized Runs - Trajectory 1	I-10
I.19. Comparison of Single Initialized Runs - Trajectory 1 w/ Payload	I-11
I.20. Comparison of Single Initialized Runs - Trajectory 1 w/ Payload	I-11
I.21. Comparison of Single Initialized Runs - Trajectory 1 w/ Payload	I-12
I.22. Comparison of Single Initialized Runs - Trajectory 6	I-12
I.23. Comparison of Single Initialized Runs - Trajectory 6	I-13
I.24. Comparison of Single Initialized Runs - Trajectory 6	I-13
J.1. Comparison of Initialized Runs After Learning - Trajectory 2	J-2
J.2. Comparison of Initialized Runs After Learning - Trajectory 2	J-2
J.3. Comparison of Initialized Runs After Learning - Trajectory 2	J-3
J.4. Comparison of Initialized Runs After Learning - Trajectory 3	J-3
J.5. Comparison of Initialized Runs After Learning - Trajectory 3	J-4
J.6. Comparison of Initialized Runs After Learning - Trajectory 3	J-4
J.7. Comparison of Initialized Runs After Learning - Trajectory 3 w/ Payload	J-5
J.8. Comparison of Initialized Runs After Learning - Trajectory 3 w/ Payload	J-5
J.9. Comparison of Initialized Runs After Learning - Trajectory 3 w/ Payload	J-6

Figure	Page
J.10. Comparison of Initialized Runs After Learning - Trajectory 4	J-6
J.11. Comparison of Initialized Runs After Learning - Trajectory 4	J-7
J.12. Comparison of Initialized Runs After Learning - Trajectory 4	J-7
J.13. Comparison of Initialized Runs After Learning - Trajectory 5	J-8
J.14. Comparison of Initialized Runs After Learning - Trajectory 5	J-8
J.15. Comparison of Initialized Runs After Learning - Trajectory 5	J-9
J.16. Comparison of Initialized Runs After Learning - Trajectory 1	J-9
J.17. Comparison of Initialized Runs After Learning - Trajectory 1	J-10
J.18. Comparison of Initialized Runs After Learning - Trajectory 1	J-10
J.19. Comparison of Initialized Runs After Learning - Trajectory 1 w/ Payload	J-11
J.20. Comparison of Initialized Runs After Learning - Trajectory 1 w/ Payload	J-11
J.21. Comparison of Initialized Runs After Learning - Trajectory 1 w/ Payload	J-12
J.22. Comparison of Initialized Runs After Learning - Trajectory 6	J-12
J.23. Comparison of Initialized Runs After Learning - Trajectory 6	J-13
J.24. Comparison of Initialized Runs After Learning - Trajectory 6	J-13
K.1. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 2 .	K-2
K.2. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 2 .	K-2
K.3. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 2 .	K-3
K.4. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3 .	K-3
K.5. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3 .	K-4
K.6. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3 .	K-4
K.7. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3 w/ Payload	K-5
K.8. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3 w/ Payload	K-5
K.9. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3 w/ Payload	K-6

Figure	Page
K.10.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 4 .	K-6
K.11.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 4 .	K-7
K.12.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 4 .	K-7
K.13.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	K-8
K.14.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	K-8
K.15.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5 .	K-9
K.16.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 .	K-9
K.17.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 .	K-10
K.18.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 .	K-10
K.19.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 w/ Payload	K-11
K.20.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 w/ Payload	K-11
K.21.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 w/ Payload	K-12
K.22.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 6 .	K-12
K.23.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 6 .	K-13
K.24.Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 6 .	K-13
P.1. The Effects of Trajectory 0 Tuning Parameters on Trajectories 2 and 3	P-2
P.2. The Effects of Trajectory 0 Tuning Parameters on Trajectories 2 and 3	P-2
P.3. The Effects of Trajectory 0 Tuning Parameters on Trajectories 2 and 3	P-3
P.4. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5	P-3
P.5. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5	P-4
P.6. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5	P-4
P.7. Payload Effects on Trajectory 0 Tuning Parameters	P-5
P.8. Payload Effects on Trajectory 0 Tuning Parameters	P-5
P.9. Payload Effects on Trajectory 0 Tuning Parameters	P-6

Figure	Page
P.10. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6	P-6
P.11. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6	P-7
P.12. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6	P-7
P.13. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec	P-8
P.14. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec	P-8
P.15. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec	P-9
Q.1. Comparison of Tarokh and Seraji Controllers - Trajectory 0	Q-2
Q.2. Comparison of Tarokh and Seraji Controllers - Trajectory 0	Q-2
Q.3. Comparison of Tarokh and Seraji Controllers - Trajectory 0	Q-3
Q.4. Comparison of Tarokh and Seraji Controllers - Trajectory 0 w/ payload	Q-3
Q.5. Comparison of Tarokh and Seraji Controllers - Trajectory 0 w/ Payload	Q-4
Q.6. Comparison of Tarokh and Seraji Controllers - Trajectory 0 w/ Payload	Q-4
Q.7. Comparison of Tarokh and Seraji Controllers - Trajectory 2	Q-5
Q.8. Comparison of Tarokh and Seraji Controllers - Trajectory 2	Q-5
Q.9. Comparison of Tarokh and Seraji Controllers - Trajectory 2	Q-6
Q.10. Comparison of Tarokh and Seraji Controllers - Trajectory 3	Q-6
Q.11. Comparison of Tarokh and Seraji Controllers - Trajectory 3	Q-7
Q.12. Comparison of Tarokh and Seraji Controllers - Trajectory 3	Q-7
Q.13. Comparison of Tarokh and Seraji Controllers - Trajectory 3 w/ Payload	Q-8
Q.14. Comparison of Tarokh and Seraji Controllers - Trajectory 3 w/ Payload	Q-8
Q.15. Comparison of Tarokh and Seraji Controllers - Trajectory 3 w/ Payload	Q-9
Q.16. Comparison of Tarokh and Seraji Controllers - Trajectory 4	Q-9
Q.17. Comparison of Tarokh and Seraji Controllers - Trajectory 4	Q-10
Q.18. Comparison of Tarokh and Seraji Controllers - Trajectory 4	Q-10
Q.19. Comparison of Tarokh and Seraji Controllers - Trajectory 5	Q-11
Q.20. Comparison of Tarokh and Seraji Controllers - Trajectory 5	Q-11
Q.21. Comparison of Tarokh and Seraji Controllers - Trajectory 5	Q-12

Figure	Page
Q.22.Comparison of Tarokh and Seraji Controllers - Trajectory 1	Q-12
Q.23.Comparison of Tarokh and Seraji Controllers - Trajectory 1	Q-13
Q.24.Comparison of Tarokh and Seraji Controllers - Trajectory 1	Q-13
Q.25.Comparison of Tarokh and Seraji Controllers - Trajectory 1 w/ Payload	Q-14
Q.26.Comparison of Tarokh and Seraji Controllers - Trajectory 1 w/ Payload	Q-14
Q.27.Comparison of Tarokh and Seraji Controllers - Trajectory 1 w/ Payload	Q-15
Q.28.Comparison of Tarokh and Seraji Controllers - Trajectory 6	Q-15
Q.29.Comparison of Tarokh and Seraji Controllers - Trajectory 6	Q-16
Q.30.Comparison of Tarokh and Seraji Controllers - Trajectory 6	Q-16
R.1. Comparison of MBAIC and SMBC Controllers - Trajectory 0	R-2
R.2. Comparison of MBAIC and SMBC Controllers - Trajectory 0	R-2
R.3. Comparison of MBAIC and SMBC Controllers - Trajectory 0	R-3
R.4. Comparison of MBAIC and SMBC Controllers - Trajectory 0 w/ payload	R-3
R.5. Comparison of MBAIC and SMBC Controllers - Trajectory 0 w/ Payload	R-4
R.6. Comparison of MBAIC and SMBC Controllers - Trajectory 0 w/ Payload	R-4
R.7. Comparison of MBAIC and SMBC Controllers - Trajectory 2	R-5
R.8. Comparison of MBAIC and SMBC Controllers - Trajectory 2	R-5
R.9. Comparison of MBAIC and SMBC Controllers - Trajectory 2	R-6
R.10.Comparison of MBAIC and SMBC Controllers - Trajectory 3	R-6
R.11.Comparison of MBAIC and SMBC Controllers - Trajectory 3	R-7
R.12.Comparison of MBAIC and SMBC Controllers - Trajectory 3	R-7
R.13.Comparison of MBAIC and SMBC Controllers - Trajectory 3 w/ Payload	R-8
R.14.Comparison of MBAIC and SMBC Controllers - Trajectory 3 w/ Payload	R-8
R.15.Comparison of MBAIC and SMBC Controllers - Trajectory 3 w/ Payload	R-9
R.16.Comparison of MBAIC and SMBC Controllers - Trajectory 4	R-9
R.17.Comparison of MBAIC and SMBC Controllers - Trajectory 4	R-10
R.18.Comparison of MBAIC and SMBC Controllers - Trajectory 4	R-10

Figure	Page
R.19.Comparison of MBAIC and SMBC Controllers - Trajectory 5	R-11
R.20.Comparison of MBAIC and SMBC Controllers - Trajectory 5	R-11
R.21.Comparison of MBAIC and SMBC Controllers - Trajectory 5	R-12
R.22.Comparison of MBAIC and SMBC Controllers - Trajectory 1	R-12
R.23.Comparison of MBAIC and SMBC Controllers - Trajectory 1	R-13
R.24.Comparison of MBAIC and SMBC Controllers - Trajectory 1	R-13
R.25.Comparison of MBAIC and SMBC Controllers - Trajectory 1 w/ Payload	R-14
R.26.Comparison of MBAIC and SMBC Controllers - Trajectory 1 w/ Payload	R-14
R.27.Comparison of MBAIC and SMBC Controllers - Trajectory 1 w/ Payload	R-15
R.28.Comparison of MBAIC and SMBC Controllers - Trajectory 6	R-15
R.29.Comparison of MBAIC and SMBC Controllers - Trajectory 6	R-16
R.30.Comparison of MBAIC and SMBC Controllers - Trajectory 6	R-16
S.1. Comparison of MBAIC and AMBC Controllers - Trajectory 2	S-2
S.2. Comparison of MBAIC and AMBC Controllers - Trajectory 2	S-2
S.3. Comparison of MBAIC and AMBC Controllers - Trajectory 2	S-3
S.4. Comparison of MBAIC and AMBC Controllers - Trajectory 3	S-3
S.5. Comparison of MBAIC and AMBC Controllers - Trajectory 3	S-4
S.6. Comparison of MBAIC and AMBC Controllers - Trajectory 3	S-4
S.7. Comparison of MBAIC and AMBC Controllers - Trajectory 3 w/ Payload	S-5
S.8. Comparison of MBAIC and AMBC Controllers - Trajectory 3 w/ Payload	S-5
S.9. Comparison of MBAIC and AMBC Controllers - Trajectory 3 w/ Payload	S-6
S.10.Comparison of MBAIC and AMBC Controllers - Trajectory 4	S-6
S.11.Comparison of MBAIC and AMBC Controllers - Trajectory 4	S-7
S.12.Comparison of MBAIC and AMBC Controllers - Trajectory 4	S-7
S.13.Comparison of MBAIC and AMBC Controllers - Trajectory 5	S-8
S.14.Comparison of MBAIC and AMBC Controllers - Trajectory 5	S-8
S.15.Comparison of MBAIC and AMBC Controllers - Trajectory 5	S-9

Figure	Page
S.16. Comparison of MBAIC and AMBC Controllers - Trajectory 1	S-9
S.17. Comparison of MBAIC and AMBC Controllers - Trajectory 1	S-10
S.18. Comparison of MBAIC and AMBC Controllers - Trajectory 1	S-10
S.19. Comparison of MBAIC and AMBC Controllers - Trajectory 1 w/ Payload	S-11
S.20. Comparison of MBAIC and AMBC Controllers - Trajectory 1 w/ Payload	S-11
S.21. Comparison of MBAIC and AMBC Controllers - Trajectory 1 w/ Payload	S-12
S.22. Comparison of MBAIC and AMBC Controllers - Trajectory 6	S-12
S.23. Comparison of MBAIC and AMBC Controllers - Trajectory 6	S-13
S.24. Comparison of MBAIC and AMBC Controllers - Trajectory 6	S-13

List of Tables

Table	Page
2.1. Seraji's Design Parameters	2-11
2.2. Tarokh's Design Parameters for Simulations and Experimental Results	2-14
3.1. Test Trajectories	3-5
3.2. Learned Parameter Values - Trajectory 1	3-13
4.1. Tarokh's Design Parameters	4-1
4.2. Design Parameters Selected During Tuning Tarokh's Algorithm	4-10
4.3. Seraji's Design Parameters	4-13
5.1. MBAIC Tuning Parameters	5-1
L.1. Learned 13-Parameter Values - Trajectory 1	L-1
L.2. Learned 13-Parameter Values - Trajectory 3	L-2
L.3. Learned 13-Parameter Values - Trajectories 2 and 4	L-2
L.4. Learned 13-Parameter Values - Trajectories 5 and 6	L-3
M.1. Learned 16-Parameter Values - Trajectory 1	M-1
M.2. Learned 16-Parameter Values - Trajectory 3	M-2
M.3. Learned 16-Parameter Values - Trajectories 2 and 4	M-2
M.4. Learned 16-Parameter Values - Trajectories 5 and 6	M-3
N.1. Learned 19-Parameter Values - Trajectory 1	N-1
N.2. Learned 19-Parameter Values - Trajectory 3	N-2
N.3. Learned 19-Parameter Values - Trajectories 2 and 4	N-3
N.4. Learned 19-Parameter Values - Trajectories 5 and 6	N-4
O.1. Learned Parameter Values - Trajectory 1	O-1

Table	Page
O.2. Learned Parameter Values - Trajectory 1 with payload	O-2
O.3. Learned Parameter Values - Trajectory 2	O-3
O.4. Learned Parameter Values - Trajectory 3	O-4
O.5. Learned Parameter Values - Trajectory 3 with payload	O-5
O.6. Learned Parameter Values - Trajectory 4	O-6
O.7. Learned Parameter Values - Trajectory 5	O-7
O.8. Learned Parameter Values - Trajectory 6	O-8

Abstract

Two robust model-based controllers and two decentralized adaptive controllers are experimentally evaluated. Algorithm evaluation is motivated by the need for controllers with good high speed tracking under varying payload conditions. The test case is a PUMA-560 robotic manipulator operating over the standard test suite.

The model-based controllers are made robust by the addition of either an auxiliary input term or an adaptive feedforward compensator based on Lyapunov Theory. The model-based auxiliary input controller (MBAIC) adapts the gain matrices used in computing an additional torque to be combined with model-based feedforward and PD feedback torques. The adaptive model-based controllers adapt the assessment of the manipulator parameters used in calculating feedforward torque.

The decentralized adaptive controllers are based on either Lyapunov stability or Popov hyperstability. These controllers calculate feedforward, feedback, and auxiliary torques based on trajectory errors and desired trajectory parameters. The gain matrices used to multiply these quantities are adapted. These auxiliary torque components are identical to those used in the MBAICs.

Experimental evaluation provides insight into the potential and limitations of each method. The decentralized digital adaptive control algorithms produce an unsatisfactory tracking response. Both model-based control techniques improve the manipulator's tracking response.

INVESTIGATION OF ADAPTIVE CONTROLLERS FOR PUMA TRAJECTORY TRACKING

I. Introduction

1.1 Motivation

The United States Air Force proposes using robotic manipulators to replace humans in hazardous or hostile environments. Robotic telepresence and aircraft refueling applications will require a manipulator with the ability to emulate the performance of a human arm. These manipulators must be able to track high-speed trajectories under a variety of payload conditions. Research on alternative robotic control methods must be expanded. Current control methods need to be more fully explored and tuned to derive optimum performance from specific manipulators.

1.2 Problem Statement

A robot is commanded to follow a desired trajectory by its control algorithm. Current industrial applications use controllers employing Proportional Derivative (PD) or Proportional Integral Derivative (PID) methods for applications where payloads and trajectories are fixed [6]. These controllers, which feed back the measured errors in an effort to compensate for disturbances produced by unmodeled robot dynamics, are designed based on the assumption that the robot dynamics are modeled by a series of linear second-order systems [8]. PD and PID controllers perform adequately when the disturbances are small. Robot's dynamics models can not be assumed linear under conditions of variable manipulator speed. Varying payloads violate the assumption of a constant model [15]. Under either of these two conditions, precomputed constants for PD or PID controllers become invalid and cannot provide the desired level of trajectory tracking accuracy [15].

1.3 Research Objective

An area of active research is the development of adaptive control algorithms that account for the nonlinear dynamics inherent in a robotic manipulator [12, 13, 17, 19]. A basic form of adaptive control is the model-based approach. In model-based control, the controller adapts to changes in robot configurations (speed and trajectory). Experimental evaluations have demonstrated the potential for model-based techniques to improve high-speed trajectory tracking accuracy [1, 7, 9]. Unknown payload variations reduce the effects of these control schemes [9]. Control algorithms that incorporate knowledge of manipulator system dynamics and are robust and/or adapt to variations in those dynamics caused by model inaccuracies, payload variation, and environmental interaction will be required [11]. Control algorithms that are independent of the dynamics model are also under consideration [20, 22, 23]. These specific algorithms are decentralized or independent of joint-to-joint interaction. The objective of this research is to continue evaluation of an adaptive model-based control algorithm and to investigate decentralized digital adaptive control techniques.

1.4 Method of Approach

As part of the Air Force Institute of Technology (AFIT) Robotics Laboratory's Gross Motion Control Program, this thesis investigated three different types of control algorithms: Adaptive Model-Based Control (AMBC), decentralized adaptive control, and Model-Based Auxiliary Input Control (MBAIC).

The comparisons and experimental evaluations performed during this thesis were conducted under the AFIT Robotic Control Algorithm Development and Evaluation (ARCADE) environment. ARCADE is hosted on a VAXstation III and has both serial and parallel connections to the PUMA computer bus. The ARCADE environment is discussed further in Chapter 3. The PUMA-560 manipulator environment is chosen because any algorithm that performs well on a PUMA should work even better on a more modern design [11]. Plots of the trajectory errors will be produced to display the error data collected.

The AMBC controller is based on the robot's dynamics model. The strength of an AMBC controller is its ability to alter the robot's manipulator parameters to improve tracking response. Repeating the same trajectory several times, and allowing the controller to retain previously computed parameters, is referred to as learning. A nominal approximation of these parameters may be used to start the controller (the controller would be initialized), or the parameters could be set to zero (uninitialized). The AMBC controller had already been implemented and evaluated in-house [12, 13]. This software was modified to include static friction parameters in an attempt to improve initial and endpoint errors. The dynamics model was rearranged to test the importance of friction compensation. All three implementations of the AMBC controller were evaluated for single-run trajectories and learning tests using initialized and uninitialized parameters.

Decentralized adaptive control algorithms are independent of the robot dynamics model. All matrices used in calculating control torques are diagonal to eliminate inter-joint dependence (thus the term "decentralized"). Seraji's Lyapunov-based decentralized digital adaptive controller was implemented by Leahy [20, 10]. New software subroutines, based loosely on this code, were written to test Tarokh's hyperstability-based algorithm. These controllers use the products of adaptive matrices and desired trajectory components or position errors to calculate feedback, feedforward, and auxiliary torques. After tuning, Tarokh's controller was evaluated for trajectory and payload independence. The test results were also compared to Leahy's work on Seraji's controller [10].

Model-Based Auxiliary Input Controllers combine the dynamics model of the AMBC algorithm with the Auxiliary Torque of decentralized adaptive controller. The model-based feedforward torque is computed from desired trajectory components. The controller uses PD feedback torques. Leahy implemented and tested a Seraji-Based MBAIC [10]. A Tarokh-based MBAIC was constructed, tuned, tested, and compared to Leahy's work.

All tests were conducted over a proposed set of standard trajectories that excite all the PUMA dynamics and allow a rigorous test of algorithm performance [13]. All of the trajectories were traversed under no-load conditions. Motor saturation constraints limited the range of payload testing available. Some of the test trajectories were run with a 2 kilogram brass disk attached to the robot's end effector. The mounting of this simple pay-

load on the end-effector was an adequate approximation of the very real task of robot tool changing and covered a large subset of the payload adaptation tasks required for a robot to emulate human arm performance. The payload was heavy enough to produce a significant degradation in algorithm performance without driving the motors into saturation [10].

1.5 Accomplishments

Evaluations of an existing 16-parameter Adaptive Model-Based Controller (AMBC) [13] demonstrated that the addition of three static friction parameters would not improve system response to a wide variety of trajectory and payload conditions. A 13-parameter AMBC controller that concentrated heavily on friction compensation did not provide sufficient tracking capability but showed the inability of an AMBC to maintain tracking accuracy when it adapts parameters with incorrect signs.

Examination of two decentralized adaptive control algorithms [20, 22] revealed their limitations and unsuitability for gross motion control applications. This was AFIT's initial examination of Tarokh's decentralized adaptive controller and the second glance at Seraji's. However, Model-Based Auxiliary Input Controllers showed promise. These controllers are based on the auxiliary torque components derived from the decentralized controllers and an SMBC controller. The MBAIC controller is simple to implement, relatively easy to tune, and provide tracking comparable to AMBC controllers. The MBAIC needs to be examined further to investigate potential uses.

1.6 Organization

The remainder of this thesis is presented in five chapters. Chapter 2 presents a literature review of adaptive model-based control schemes and decentralized adaptive control algorithms currently under investigation at the Air Force Institute of Technology. Chapter 3 discusses the tuning, testing, and analysis of the adaptive model-based control algorithm. Chapter 4 covers the tuning, testing, and analysis of the decentralized adaptive control algorithms. Chapter 5 covers the tuning, testing, and analysis of the Model-Based Auxiliary Input Control algorithms. Finally, Chapter 6 presents conclusions and recommendations for further research.

II. Literature Review

2.1 Introduction

This chapter is divided into two main sections. First, model-based robotic control is discussed as a prelude to examining an adaptive model-based control algorithm that has demonstrated its effectiveness in previous in-house evaluations [11, 12, 13]. The second portion of this chapter explores two decentralized adaptive control algorithms [20, 22, 23]. One controller is based on Lyapunov stability criteria while the other is formulated using Popov hyperstability theory. These decentralized algorithms are independent of robot dynamics models.

2.2 Adaptive Model-Based Control Algorithm

2.2.1 Model-Based Control. A dynamics model of the robot manipulator must be developed in order to use model-based control techniques. Ignoring motor dynamics the equations of motion may be written in vector form as [4, 18]:

$$\tau(t) = \mathbf{H}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta})\dot{\theta} + \mathbf{G}(\theta) \quad (2.1)$$

where:

n is the number of degrees of freedom of the robot manipulator

$\theta, \dot{\theta}, \ddot{\theta}$ are n -dimensional vectors of joint positions, velocities, and accelerations

$\tau_m(t)$ is an n -dimensional vector of joint motor torques

$\mathbf{H}(\theta)$ is the $n \times n$ symmetric positive-definite inertia matrix

$\mathbf{C}(\theta, \dot{\theta})\dot{\theta}$ is an n -dimensional vector of centripetal and Coriolis torques

$\mathbf{G}(\theta)$ is an n -dimensional vector of gravity torques

The end result of model-based control is to obtain an equation of the form:

$$\tau(t) = \tau_{ff} + \tau_{fb} \quad (2.2)$$

with:

τ_{ff} = feedforward compensation torques

τ_{fb} = feedback compensation torques

The right-hand side of Equation (2.1) develops the feedforward torques which may be expressed by:

$$\tau_{ff} = \hat{H}(\theta)\ddot{\theta} + \hat{C}(\theta, \dot{\theta})\dot{\theta} + \hat{G}(\theta) \quad (2.3)$$

where $\hat{\cdot}$ denotes modeled values. The feedforward torque may also be denoted by a regressor formulation which defines the torque as a function of a linear parameter vector \hat{a} :

$$\tau_{ff} = Y(\theta, \dot{\theta}, \ddot{\theta})\hat{a} \quad (2.4)$$

The \hat{H} , \hat{C} , and \hat{G} terms are expressed in terms of the parameter vector and combined to form the regressor matrix Y . The modeling attributed to the H , C , and G matrices is transferred to the manipulator parameters.

The feedback torques are:

$$\tau_{fb} = K_v\dot{e} + K_p e \quad (2.5)$$

where

\dot{e} is the joint velocity error, $\dot{\theta}_d - \dot{\theta}$

e is the joint position error, $\theta_d - \theta$

$\theta_d, \dot{\theta}_d$ are n -dimensional vectors of desired joint positions and velocities

K_v, K_p are diagonal $n \times n$ matrices of velocity and position feedback loop gains

The ideal situation, arrived at by substituting Equations (2.3) and (2.5) into Equation (2.2), equating the result to Equation (2.1) and assuming that the modeled values are equal to the actual values, is denoted by:

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \quad (2.6)$$

Equation (2.6) states that, with the appropriate K_p and K_v values, the model-based controller uniformly rejects errors over the manipulator's entire trajectory. This equivalence assumption of modeled and actual values ignores uncertainties that arise from time variations in the robot's transfer function produced by the operating environment, component variation due to wear, and manufacturing irregularities [8].

2.2.2 Adaptive Model-Based Control. Uncertainties in the dynamics model, Equation (2.3), are passed on to the model-based controller as disturbances. The controller can maintain good high-speed tracking performance in the presence of small disturbances. Unknown payload parameters cause degradation of the controller's performance. Some adaptive control algorithms estimate full dynamics feedforward torque compensation, with unknown manipulator and payload parameters, on line. These algorithms may be supplemented by PD feedback torques. Adaptive control schemes based on Lyapunov stability theory have been developed by Sadegh and Horowitz [19], Slotine and Li [21], and others [17, 24].

For this discussion, motor dynamics were ignored to facilitate the presentation. During the actual implementation they must be considered. After defining a control law of:

$$\tau = \hat{H}(\theta)\ddot{\theta}_d + \hat{C}(\theta, \dot{\theta})\dot{\theta}_d + \hat{G}(\theta) + K_p\tilde{e} + K_v\dot{\tilde{e}} \quad (2.7)$$

$$= Y(\theta, \dot{\theta}, \ddot{\theta}_d)\hat{a} + K_p\tilde{e} + K_v\dot{\tilde{e}} \quad (2.8)$$

Slotine and Li derive an adaptive control algorithm from a Lyapunov stability analysis [21]. They used a Lyapunov function candidate of:

$$V(t) = \frac{1}{2}(\dot{e}^T H(\theta)\dot{e} + \tilde{a}^T \Gamma \tilde{a} + e^T K_p e) \quad (2.9)$$

where:

Γ is a symmetric positive-definite matrix (usually diagonal)

\hat{a} is the vector of manipulator parameter estimates

$\tilde{a} = \hat{a}(t) - a$ is the parameter estimation error vector

Differentiating Equation (2.9) and using the property of skew symmetry described in [21] to eliminate the $\frac{1}{2}\dot{\mathbf{e}}^T(\dot{\mathbf{H}} - 2\mathbf{C})\dot{\mathbf{e}}$ term yields:

$$\dot{V}(t) = \dot{\mathbf{e}}^T[\tau - \mathbf{H}(\theta)\ddot{\theta}_d - \mathbf{C}(\theta, \dot{\theta})\dot{\theta}_d - \mathbf{G}(\theta) - \mathbf{K}_p\mathbf{e}] + \tilde{\mathbf{a}}^T\Gamma\dot{\tilde{\mathbf{a}}} \quad (2.10)$$

$$= \dot{\mathbf{e}}^T[\tau - \mathbf{Y}(\theta, \dot{\theta}, \ddot{\theta}_d)\tilde{\mathbf{a}} - \mathbf{K}_p\mathbf{e}] + \tilde{\mathbf{a}}^T\Gamma\dot{\tilde{\mathbf{a}}} \quad (2.11)$$

After substituting Equation (2.7) for τ and defining:

$$\tilde{\mathbf{H}}(\theta) = \dot{\mathbf{H}}(\theta) - \mathbf{H}(\theta)$$

$$\tilde{\mathbf{C}}(\theta, \dot{\theta}) = \dot{\mathbf{C}}(\theta, \dot{\theta}) - \mathbf{C}(\theta, \dot{\theta})$$

$$\tilde{\mathbf{G}}(\theta) = \dot{\mathbf{G}}(\theta) - \mathbf{G}(\theta)$$

the derivative equation (Equation (2.10)) becomes:

$$\dot{V}(t) = -\dot{\mathbf{e}}^T\mathbf{K}_v\dot{\mathbf{e}} - \dot{\mathbf{e}}^T[\tilde{\mathbf{H}}(\theta)\ddot{\theta}_d + \tilde{\mathbf{C}}(\theta, \dot{\theta})\dot{\theta}_d + \tilde{\mathbf{G}}(\theta)] + \tilde{\mathbf{a}}^T\Gamma\dot{\tilde{\mathbf{a}}} \quad (2.12)$$

$$= -\dot{\mathbf{e}}^T\mathbf{K}_v\dot{\mathbf{e}} - \dot{\mathbf{e}}^T\mathbf{Y}(\theta, \dot{\theta}, \ddot{\theta}_d)\tilde{\mathbf{a}} + \tilde{\mathbf{a}}^T\Gamma\dot{\tilde{\mathbf{a}}} \quad (2.13)$$

Transposing all the elements in the second term of the right-hand side of Equation (2.13) and combining the last two terms leads to:

$$\dot{V}(t) = -\dot{\mathbf{e}}^T\mathbf{K}_v\dot{\mathbf{e}} + \tilde{\mathbf{a}}^T[\Gamma\dot{\tilde{\mathbf{a}}} - \mathbf{Y}^T(\theta, \dot{\theta}, \ddot{\theta}_d)\dot{\mathbf{e}}] \quad (2.14)$$

For Lyapunov stability to hold $\dot{V}(t)$ must be zero or negative [21]. Knowing that \mathbf{K}_v is defined as positive definite, then setting:

$$\Gamma\dot{\tilde{\mathbf{a}}} - \mathbf{Y}^T(\theta, \dot{\theta}, \ddot{\theta}_d)\dot{\mathbf{e}} = 0 \quad (2.15)$$

leaves:

$$\dot{V}(t) = -\dot{\mathbf{e}}^T\mathbf{K}_v\dot{\mathbf{e}} \leq 0 \quad (2.16)$$

and the system is stable.

Rewriting Equation (2.15) to solve for the unknown vector $\dot{\hat{\mathbf{a}}}$ results in:

$$\dot{\hat{\mathbf{a}}} = \Gamma^{-1} \mathbf{Y}^T(\theta, \dot{\theta}, \ddot{\theta}_d) \dot{\mathbf{e}} \quad (2.17)$$

Since \mathbf{a} is constant, $\dot{\hat{\mathbf{a}}} = \dot{\hat{\mathbf{a}}}$, and we have an equation for determining an accurate model for τ_{ff} .

2.2.3 Experimentation Algorithm. The model-based algorithm for the output torque used by Leahy and Whalen [13], adapted from Sadegh and Horowitz [19], contains an auxiliary input, τ_{ax} , plus feedforward and feedback terms. It is:

$$\tau = \tau_{ff} + \tau_{fb} + \tau_{ax} \quad (2.18)$$

The feedback portion of the control law is:

$$\tau_{fb} = \mathbf{K}_v[(\dot{\theta}_d - \dot{\theta}) + \Lambda(\theta_d - \theta)] \quad (2.19)$$

where Λ is a constant matrix with eigenvalues in the right-half complex plane. Feedback torque is related to Equation (2.16), which was Slotine and Li's expression for Lyapunov stability. Λ is selected to be $\mathbf{K}_p/\mathbf{K}_v$ in order to implement PD feedback as expressed in Equation (2.5).

The feedforward component of this equation is broken down into two parts to allow for precalculation and off-line storage of the feedforward torques that are functions of the known parameters, $\hat{\mathbf{a}}_n$, and desired joint positions, velocities, and accelerations. The divided equation is:

$$\begin{aligned} \tau_{ff} &= \mathbf{Y}(\theta_d, \dot{\theta}_d, \ddot{\theta}_d) \hat{\mathbf{a}} \\ &= \mathbf{Y}_1(\theta_d, \dot{\theta}_d, \ddot{\theta}_d) \hat{\mathbf{a}} + \mathbf{Y}_2(\theta_d, \dot{\theta}_d, \ddot{\theta}_d) \hat{\mathbf{a}}_n \end{aligned} \quad (2.20)$$

The selection of Equation (2.7) as the control law allows the cancellation of terms associated with the known manipulator parameters, leaving only the unknown parameters to be estimated by $\hat{\mathbf{a}}$. The regressor matrix does not depend on the actual manipulator

acceleration, but on the velocity and acceleration of the desired trajectory [17], which eliminates sensitivity problems that may be induced by imprecise measurements if on-line measurements of manipulator acceleration were available [21].

The regressor matrix differs from Slotine and Li's version because it is only dependent on desired trajectory components, not a combination of actual and desired trajectory attributes. The feedforward torque component based on known components, $Y_2(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)\hat{a}_n$, may be precomputed and stored off-line. The portion of the regressor associated with the unknown parameters, $Y_1(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)$, can also be precomputed. This approach leads to a reduction in the number of on-line calculations which permits the use of an increased sampling rate [16]. The main advantage of using desired trajectory quantities is that the noisy measurements of actual velocity and acceleration are avoided. Other advantages afforded by adaptive model-based algorithms are increased robustness and the capability of the manipulator to "learn" in the case of repetitive tasks.

The parameter adaptation for evaluating the unknown manipulator parameters on line was accomplished through the use of a Desired Compensation Adaptation Law (DCAL). Sadegh and Horowitz's simulations [19] and experimental results [16] show that the use of a DCAL allowed their manipulator to maintain stability and retain parameter convergence in the presence of noise and a constant adaptation signal. The adaptation law is:

$$\dot{\hat{a}} = \int \Gamma^{-1} Y^T(\theta_d, \dot{\theta}_d, \ddot{\theta}_d) [(\dot{\theta}_d - \dot{\theta}) + \Lambda(\theta_d - \theta)] \quad (2.21)$$

This adaptation equation is nearly the same as Equation (2.17), except for the addition of the weighted position error term $\Lambda(\theta_d - \theta)$. This regressor matrix, Y , which contains only desired trajectory components, can be precalculated and stored off-line. As in the case of the precomputed feedforward torques, using a regressor matrix based on desired trajectory values enhances robustness and avoids noise corruption of the error and adaptation signals.

The auxiliary torque term provides compensation for additional errors introduced by modifying the adaptive controller [19]. The auxiliary torque, when used, is evaluated by:

$$\tau_{ax} = \sigma_n |e|^2 [(\dot{\theta}_d - \dot{\theta}) + \Lambda(\theta_d - \theta)] \quad (2.22)$$

This function was set to zero [13] since the small errors anticipated from using this algorithm make this torque component insignificant.

2.2.4 Friction Models. Several authors [19, 11, 17] note the importance of friction in the manipulator dynamics model. The different types of friction are viscous, Coulomb, and static friction [8]. Viscous friction, $\tau_v(t)$, is characterized as a linear relationship between applied force and velocity. Coulomb (or running) friction, $\tau_c(t)$, has a constant amplitude with its sign dependent on the sign of the velocity. Static friction, $\tau_s(t)$, is the resistance to initial motion which vanishes with the onset of movement. The sign of the static friction term is such that the force opposes motion. These forces are defined respectively by [8]:

$$\tau_v(t) = B\dot{\theta}(t) \quad (2.23)$$

$$\tau_c(t) = \tau_c \text{sgn}(\dot{\theta}) \quad (2.24)$$

$$\tau_s(t) = \pm \tau_s|_{\dot{\theta}=0} \quad (2.25)$$

The static friction term has also been expressed as a constant times an exponential term [2, 3]. Canudas de Wit and Seront use Tustin's model to express the overall friction model as [5]:

$$\tau(\dot{\theta}) = [\alpha_0 + \alpha_1 e^{-\tau_0|\dot{\theta}|} + \alpha_2|\dot{\theta}|]\text{sgn}(\dot{\theta}) \quad (2.26)$$

The terms α_0 , α_1 , and α_2 correspond to the τ_c , τ_s , and B respectively from the previous definitions. The exponential term allows for the static friction term to be a rapidly decreasing function rather than a spike. The term τ_0 is the lubricant coefficient which is between 20 and 60 for most materials. These authors suggest using 40 for the lubrication coefficient.

Friction has a significant effect on the performance on many robotic manipulators, especially those with gear reduction. Friction compensation is a challenge because its functional dependence on the joint variables is difficult to model [17]. Leahy and Whalen state that their adaptive algorithm benefits from its ability to adapt Coulomb and viscous friction terms [12]. This adaptation is shown to reduce the tracking errors on both ends of their test trajectories where velocity is low and friction and gravity terms dominate the

dynamics. Additional friction compensation in the low velocity regions of the trajectory may further reduce tracking errors. The friction component that is unmodeled by both Sadegh and Horowitz [19], and Leahy and Whalen [13], is static friction.

2.3 Decentralized Digital Adaptive Control

2.3.1 Introduction. Conventional industrial controllers are based on independent joint controllers where each individual joint is controlled by a simple position servo-loop with predefined gains. This method is inadequate for applications involving precise tracking of fast trajectories under uncertain payload conditions [10, 20]. Control systems proposed to handle these operating conditions include model-based and performance-based methods. Performance-based methods do not involve the use of the complex mathematical manipulator dynamics model in the formulation of the control law.

2.3.2 Lyapunov-Based Decentralized Adaptive Control. Seraji proposed a simple Lyapunov-based decentralized adaptive independent joint control technique [20]. This technique does not require any knowledge of the manipulator dynamics model, model parameters, or payload parameters.

The coupled manipulator dynamics model needs to be massaged to develop a decentralized control scheme. Seraji's dynamics model is [20]:

$$\mathbf{T} = \mathbf{H}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) + \mathbf{S}(\dot{\boldsymbol{\theta}}) \quad (2.27)$$

where:

$\mathbf{T}(t)$ is the $n \times 1$ vector of applied joint torques

$\mathbf{S}(\dot{\boldsymbol{\theta}})$ is the $n \times 1$ frictional torque vector

This model can be broken down into scalar differential equations of the form:

$$T_i(t) = h_{ii}(\boldsymbol{\theta})\ddot{\theta}_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n h_{ij}(\boldsymbol{\theta})\ddot{\theta}_j(t) + c_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\theta}_i + g_i(\boldsymbol{\theta}) + s_i(\dot{\boldsymbol{\theta}}) \quad (2.28)$$

where i denotes the i th element through the n th joint and the lower case letters h, c, g, s are the individual joint variables of the quantities represented by the same upper case letters. This model is equivalent to:

$$T_i(t) = h_{ii}(\theta)\ddot{\theta}_i(t) + d_i(\theta, \dot{\theta}, \ddot{\theta}) \quad (2.29)$$

where:

$$d_i(\theta, \dot{\theta}, \ddot{\theta}) = \sum_{\substack{j=1 \\ j \neq i}}^n h_{ij}(\theta)\ddot{\theta}_j(t) + c_i(\theta, \dot{\theta})\dot{\theta}_i + g_i(\theta) + s_i(\dot{\theta}) \quad (2.30)$$

Now each joint is modeled by an input-output dynamic equation with the joint torque T_i as the input, the joint angle θ_i as the output, and d_i treated as a disturbance torque. From the system viewpoint, the term d_i represents the coupling between the i th joint and other joints and nonlinear dynamics. The joint control laws are restricted to be decentralized, meaning that each controller operates on only its own joint with no interchange of information between joints. In other words, the decentralized manipulator control problem is to design a set of independent joint controllers in which the i th controller generates the joint torque $T_i(t)$ by acting only on the joint angle trajectory $\theta_i(t)$ and the desired trajectory $\theta_d(t)$, and ensures that $\theta_i(t)$ tracks $\theta_d(t)$ asymptotically. Seraji's proposed decentralized controller is [20]:

$$T_i(t) = f_i(t) + [k_{i0}(t)e_i(t) + k_{i1}(t)\dot{e}_i(t)] + [q_{i0}(t)\theta_{di}(t) + q_{i1}(t)\dot{\theta}_{di}(t) + q_{i2}(t)\ddot{\theta}_{di}(t)] \quad (2.31)$$

where, for each individual joint:

$\theta_{di}(t)$ is the desired trajectory,

$\theta_i(t)$ is the joint angle trajectory, and

$e_i(t)$ is the joint tracking error, $\theta_{di}(t) - \theta_i(t)$.

This control law has three distinct components:

1. The auxiliary signal, $f_i(t)$, partially compensates for $d_i(t)$ and improves tracking

2. The term $[k_{i0}(t)e_i(t) + k_{i1}(t)\dot{e}_i(t)]$ is the adaptive position-velocity feedback controller. The gains, $k_{i0}(t)$ and $k_{i1}(t)$, are adjustable.
3. The term $[q_{i0}(t)\theta_{di}(t) + q_{i1}(t)\dot{\theta}_{di}(t) + q_{i2}(t)\ddot{\theta}_{di}(t)]$ is the adaptive position-velocity-accelerator feedforward controller. Its gains, $q_{i0}(t)$, $q_{i1}(t)$, and $q_{i2}(t)$, are also adjustable.

Equation (2.31) can be rewritten as:

$$T_i(t) = f_i(t) + \sum_{j=0}^1 k_{ij}(t)e_i^{(j)}(t) + \sum_{j=0}^2 q_{ij}(t)\theta_{di}^{(j)}(t) \quad (2.32)$$

where the superscripts denote derivatives with respect to time. The controller adaptation laws are based on the weighted error $r_i(t)$ as follows:

weighted error:

$$r_i(t) = w_{pi}e_i(t) + w_{vi}\dot{e}_i(t) \quad (2.33)$$

auxiliary signal:

$$f_i(t) = f_i(0) + \delta_i \int_0^t r_i(t)dt + p_i r_i(t) \quad (2.34)$$

feedback gains:

$$k_{ij}(t) = k_{ij}(0) + \alpha_{ij} \int_0^t r_i(t)e_i^{(j)}(t)dt + \beta_{ij}r_i(t)e_i^{(j)}(t) \quad j = 0, 1 \quad (2.35)$$

feedforward gains:

$$q_{ij}(t) = q_{ij}(0) + \gamma_{ij} \int_0^t r_i(t)\theta_{di}^{(j)}(t)dt + \lambda_{ij}r_i(t)\theta_{di}^{(j)}(t) \quad j = 0, 1, 2 \quad (2.36)$$

where $\{\delta_i, \alpha_{ij}, \gamma_{ij}\}$ are any positive scalar integral adaptation gains, $\{p_i, \beta_{ij}, \lambda_{ij}\}$ are zero or any positive scalar proportional adaptation gains, and $\{w_{pi}, w_{vi}\}$ are positive scalar weighting factors demonstrating the relative magnitudes of the position and velocity errors used in forming the weighted error.

This decentralized controller, consisting of several independent joint controllers, is purported to have advantages over a single centralized controller for the entire manipu-

lator [20]. The decentralized controller is more computationally efficient. For an n -joint manipulator, the decentralized controller computes $6n$ adaptive gains each sample period versus $5n^2 + n$ adaptive gain computations for a centralized controller. System performance could be further enhanced by employing n simple and fast microprocessors for parallel processing and distributed computing. A second attribute of this control scheme is reliability and fault tolerance. Erroneous joint position readings from a joint encoder would affect the entire structure of a centralized controller but only one joint of the decentralized system. A limitation of the decentralized controller is the possibility of tracking errors caused by the absence of complete inter-joint coupling compensation. The decentralized adaptive control algorithm accepts this deficiency as part of the trade-off in attaining computational simplicity, fault tolerance, and robustness.

Seraji's experiments were conducted on a PUMA-560 industrial robot over a single three joint trajectory from $(0^\circ, 0^\circ, 0^\circ)$ to $(60^\circ, -60^\circ, 60^\circ)$ in 3 seconds [20]. His experimental results did not address the issues of trajectory or payload independence using the decentralized control algorithm. Seraji's design parameters, contained in Table 2.1, reduced the control torque to:

$$\mathbf{T}(t) = \mathbf{f}(t) + \mathbf{k}_0(t)\mathbf{e}(t) + \mathbf{k}_1(t)\dot{\mathbf{e}}(t) \quad (2.37)$$

which uses only the auxiliary and feedback torque components available. The proportional pieces of the \mathbf{f} , \mathbf{k}_0 , and \mathbf{k}_1 adaptive matrices, \mathbf{p}_i , $\beta_0 \mathbf{r}_i$, and $\beta_1 \mathbf{r}_i \dot{\mathbf{e}}$ respectively, were also

Table 2.1. Seraji's Design Parameters

Auxiliary terms:	$\delta_i = [50, 50, 50]$	$p_i = [0, 0, 0]$
Feedback terms:	$\alpha_{i0} = [100, 100, 100]$	$\beta_{i0} = [0, 0, 0]$
	$\alpha_{i1} = [800, 800, 800]$	$\beta_{i1} = [0, 0, 0]$
Feedforward terms:	$\gamma_{i0} = [0, 0, 0]$	$\lambda_{i0} = [0, 0, 0]$
	$\gamma_{i1} = [0, 0, 0]$	$\lambda_{i1} = [0, 0, 0]$
	$\gamma_{i2} = [0, 0, 0]$	$\lambda_{i2} = [0, 0, 0]$
Weighted error terms:	$w_{pi} = [30, 40, 12]$	$w_{vi} = [20, 20, 4]$

omitted. Leahy [10] further evaluated this algorithm using Seraji's design parameters and initial conditions over a trajectory of $(-50^\circ, -135^\circ, 135^\circ)$ to $(45^\circ, -85^\circ, 30^\circ)$ in 1.5 seconds to excite all of the PUMA's dynamics. Manipulator performance was degraded. Leahy concluded that this control scheme's performance is heavily dependent on accurate initial compensation of static dynamic forces and the position gain adaptation properties of the algorithm are minimal.

Tarokh developed another decentralized control scheme based on Popov hyperstability theory. Hyperstability theory is used to enhance design flexibility. Tarokh's controller is discussed in the next section.

2.3.3 Discrete-Time Adaptive Control. Tarokh developed a discrete-time adaptive control algorithm employing Popov hyperstability [22, 23]. The dynamics model, similar to Equation (2.27), is written as:

$$\mathbf{u}(t) = \mathbf{B}_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, m)\ddot{\boldsymbol{\theta}} + \mathbf{B}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, m)\dot{\boldsymbol{\theta}} + \mathbf{B}_0(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, m)\boldsymbol{\theta} \quad (2.38)$$

where $\mathbf{u}(t)$ is the $n \times 1$ torque vector and $\mathbf{B}_2(\cdot)$, $\mathbf{B}_1(\cdot)$, and $\mathbf{B}_0(\cdot)$ are matrices with elements that are nonlinear functions of payload mass, $m(t)$, and manipulator joint position and velocity.

The dynamics model can be further reduced to Equation (2.39) by realizing that \mathbf{B}_2 , \mathbf{B}_1 , and \mathbf{B}_0 are time varying matrices and then discretized to form Equation (2.40) with a sample period T :

$$\mathbf{u}(t) = \mathbf{B}_2(t)\ddot{\boldsymbol{\theta}}(t) + \mathbf{B}_1(t)\dot{\boldsymbol{\theta}}(t) + \mathbf{B}_0(t)\boldsymbol{\theta}(t) \quad (2.39)$$

$$\mathbf{u}(k) = \mathbf{A}_2(k, T)\boldsymbol{\theta}(k-2) + \mathbf{A}_1(k, T)\boldsymbol{\theta}(k-1) + \mathbf{A}_0(k, T)\boldsymbol{\theta}(k) \quad (2.40)$$

Assuming a constant sample period and using a discretized joint error vector definition of $\boldsymbol{\theta}_e(k) = \boldsymbol{\theta}_d(k) - \boldsymbol{\theta}(k)$ allows Equation (2.40) to take on the form:

$$\begin{aligned} \mathbf{A}_0(k)\boldsymbol{\theta}_e(k) = & -\mathbf{u}(k) - \mathbf{A}_1(k)\boldsymbol{\theta}_e(k-1) - \mathbf{A}_2(k)\boldsymbol{\theta}_e(k-2) + \mathbf{A}_0(k)\boldsymbol{\theta}_d(k) \\ & + \mathbf{A}_1(k)\boldsymbol{\theta}_d(k-1) + \mathbf{A}_2(k)\boldsymbol{\theta}_d(k-2) \end{aligned} \quad (2.41)$$

Equation (2.41) suggests that, in order to influence the joint angle error θ_e completely, you need a control law of the general form:

$$\begin{aligned} u(k) = & P_1(k)\theta_e(k-1) + P_2(k)\theta_e(k-2) + Q_0(k)\theta_d(k) + Q_1(k)\theta_d(k-1) \\ & + Q_2(k)\theta_d(k-2) + \eta(k) \end{aligned} \quad (2.42)$$

where $P_1(k)$ and $P_2(k)$ are time-varying matrices which develop the feedback torque based upon joint angle errors and $Q_0(k)$, $Q_1(k)$, and $Q_2(k)$ are time-varying matrices which, when multiplied by the desired trajectory positions, produce the feedforward torques. The last quantity, $\eta(k)$, is an auxiliary signal added to enhance adaptation.

Tarokh utilized Popov hyperstability to ensure that the adaptation error asymptotically approached zero. A complete derivation is in [22, 23]. Stability conditions were satisfied by using proportional plus integral adaptation laws to derive the gain matrices for the i th joint. The matrices needed to compute the control torques are developed from:

$$\begin{aligned} P_{1i}(k) = & P_{1i}(k-1) + \hat{\theta}_{ei}(k)\theta_{ei}(k-1)E_{1Pi} \\ & + \hat{\theta}_{ei}(k-1)\theta_{ei}(k-2)[E_{1Ii} - E_{1Pi}] \end{aligned} \quad (2.43)$$

$$\begin{aligned} P_{2i}(k) = & P_{2i}(k-1) + \hat{\theta}_{ei}(k)\theta_{ei}(k-2)E_{2Pi} \\ & + \hat{\theta}_{ei}(k-1)\theta_{ei}(k-3)[E_{2Ii} - E_{2Pi}] \end{aligned} \quad (2.44)$$

$$\begin{aligned} Q_{0i}(k) = & Q_{0i}(k-1) + \hat{\theta}_{ei}(k)\theta_{di}(k)F_{0Pi} \\ & + \hat{\theta}_{ei}(k-1)\theta_{di}(k-1)[F_{0Ii} - F_{0Pi}] \end{aligned} \quad (2.45)$$

$$\begin{aligned} Q_{1i}(k) = & Q_{1i}(k-1) + \hat{\theta}_{ei}(k)\theta_{di}(k-1)F_{1Pi} \\ & + \hat{\theta}_{ei}(k-1)\theta_{di}(k-2)[F_{1Ii} - F_{1Pi}] \end{aligned} \quad (2.46)$$

$$\begin{aligned} Q_{2i}(k) = & Q_{2i}(k-1) + \hat{\theta}_{ei}(k)\theta_{di}(k-2)F_{2Pi} \\ & + \hat{\theta}_{ei}(k-1)\theta_{di}(k-3)[F_{2Ii} - F_{2Pi}] \end{aligned} \quad (2.47)$$

$$\eta_i(k) = \eta_i(k-1) + \beta_{Pi}\hat{\theta}_{ei}(k) + (\beta_{Ii} - \beta_{Pi})\hat{\theta}_{ei}(k-1) \quad (2.48)$$

$$\hat{\theta}_{ei}(k) = r_{2i}\theta_{ei}(k-1) + r_{3i}\theta_{ei}(k)$$

$$r_{2i} = \alpha_i\lambda_{1i}\lambda_{2i}(\lambda_{1i} + \lambda_{2i})$$

$$r_{3i} = \alpha_i(1 + \lambda_{1i}\lambda_{2i})$$

Table 2.2. Tarokh's Design Parameters for Simulations and Experimental Results

	Computer Simulation		Experimental Results	
Feedback terms:	$E_{1P} = 0I$ $E_{1I} = 3I$	$E_{2P} = 0I$ $E_{2I} = 3I$	$E_{1P} = 0I$ $E_{1I} = 2I$	$E_{2P} = 0I$ $E_{2I} = 2I$
Constants:	$\alpha_1 = 200$	$\alpha_2 = 100$	Not Given	
Eigenvalues:	$\lambda_1 = -0.7I$	$\lambda_2 = -0.7I$	$\lambda_1 = -0.5I$	$\lambda_2 = -0.5I$
Auxiliary terms:	$\beta_P = 30I$	$\beta_I = 0.02I$	$\beta_P = 50I$	$\beta_I = 0.5I$
Initial Conditions:	$\eta_1(0) = 0$ $P_1(0) = 0$	$\eta_2(0) = 0$ $P_2(0) = 0$	$\eta_1(0) = -25$ $P_1(0) = 0$	$\eta_2(0) = 3$ $P_2(0) = 0$
Sample Period:	$T_s = 1\text{msec}$		$T_s = 7\text{msec}$	

where the E and F terms are constant symmetric positive definite gain matrices, the β terms are positive scalars, and the subscripts P and I denote proportional and integral terms. The weighted joint angle errors, $\hat{\theta}_{ei}$, involve constants derived from positive constants α_i , while λ_1 and λ_2 are eigenvalues chosen from the error reference model such that their magnitudes are less than 1.

2.3.4 Simulation and Experimental Results Discussion. Like Seraji, Tarokh discussed different conditions in his computer simulation and experimental results sections. His simulation discussion evaluated a 3-second trajectory of $(90^\circ, 0^\circ)$ to $(0^\circ, 90^\circ)$ for Links 2 and 3 of a PUMA-560 using only feedback and auxiliary torques. His control law was [22]:

$$u(k) = P_1(k)\theta_e(k-1) + P_2(k)\theta_e(k-2) + \eta(k) \quad (2.49)$$

This control law uses only feedback and auxiliary compensation. Furthermore, only integral adaptation was used to compute the feedback compensation. The parameters for the computer simulation are shown in Table 2.2.

He evaluated several other trajectories, initial positions, and payload conditions and discovered tracking errors of less than 1.0 degree. The simulations also address the issue of adaptation rate. Instability could result from a long sample period.

The experimental results section concentrated on three different test scenarios summarized by:

1. Movement of Joint 2 only from 0° to 50° in 2 seconds with all other joints stationary and no payload.
2. Movement of Joint 2 only from 0° to 50° in 2 seconds with all other joints stationary and a 3 kg payload.
3. Movement of Joints 2 and 3 from $(0^\circ, 0^\circ)$ to $(50^\circ, -50^\circ)$ in 2 seconds with all other joints stationary and no payload.

The parameters for the experimental runs are different from the computer simulations. They are also shown in Table 2.2. The tracking responses of Joint 2 comparing test scenario (1) versus (2) and scenario (1) versus (3) are given in [22, 23]. These plots show that the different test conditions cause little difference in trajectory tracking.

Questions left unanswered by Tarokh which will be addressed by this effort include:

1. Is this controller applicable under a variety of harsher trajectory conditions?
2. Are the parameters chosen for one trajectory good for only that particular trajectory or several trajectories?
3. Is this control algorithm sensitive to the speed of the trajectory?

2.4 Summary

This chapter opened with a discussion of the development of an AMBC controller and detailed the implementation that was used. An overview of decentralized adaptive control followed. Tarokh's and Seraji's versions were examined.

The next chapter will discuss the results of adaptive model-based control research. It will be followed by a discussion of the decentralized digital adaptive control efforts.

III. Adaptive Model-Based Control

3.1 Introduction

In several articles Leahy and Whalen have espoused Adaptive Model-Based Control (AMBC) as a method for robotic control [12, 13]. The aim of this thesis effort was to include a third set of friction parameters in the adaptive control algorithm. These parameters addressed static friction in an effort to reduce endpoint errors, especially at the start of the trajectory. Initially the algorithms were tuned to accept this change. Several different trajectories were traversed under a variety of initial conditions. Finally the feasibility of employing only the most critical gravity and inertia terms along with the nine friction terms in the AMBC algorithm was explored.

3.2 Test Environment

The experimental evaluations performed in this effort were conducted under the AFIT Robotic Control Algorithm Development and Evaluation (ARCADE) environment. ARCADE is hosted on a VAXstation III and has both serial and parallel connections to the original PUMA computer bus. The PUMA's LSI-11/73 computer now serves strictly as a preprocessor. The angular position and motor current information is passed between the preprocessor and the VAXstation via a DRV11-J parallel interface. The parallel device driver software is supplied by the VAXlab software package. Velocity error is calculated by differencing the desired and calculated velocity signal:

$$\dot{e}(k) = \dot{q}_d(k) - [q(k) - q(k-1)]/T_s \quad (3.1)$$

Velocity noise translates into torque spikes and is the major limitation of algorithm feedback gain, and therefore tracking performance. A simple low-pass filter has been added to help reduce these noise effects. Communication restrictions, minimal processing time, and nominal clock rate resulted in a servo rate of 4.5 ms (222 Hz) for these evaluations [11].

3.3 Algorithm Implementation

The specific structure of the AMBC algorithm implemented during this research was identical to that of previous research on the PUMA. Feedforward torques are produced based on estimated and known parameters.

$$\tau = \tau_{ff} + \tau_{fb} \quad (3.2)$$

$$\tau = (Y(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)\hat{a}) + (K_v\dot{e} + K_p e) \quad (3.3)$$

Simple PD feedback torque is used with the following gains [12]:

Link #	1	2	3
K_p	640	1330	360
K_v	72	130	25

To visualize the layout of the links, think of the Puma as modeled like a human torso. Link 1 is the torso, Link 2 is the upper arm, and Link 3 is the forearm. Joints 1-3 respectively are the waist, shoulder, and elbow [6].

These torques are digitally implemented for each joint by the following set of equations:

$$\tau_{fb}(k) = K_v\dot{e}(k-1) + K_p e(k-1) \quad (3.4)$$

$$\tau_{ff}(k) = Y_1[q_d(k), \dot{q}_d(k), \ddot{q}_d(k)]\hat{a}(k) + Y_2[q_d(k), \dot{q}_d(k), \ddot{q}_d(k)]\hat{a}_n(k) \quad (3.5)$$

$$\dot{e}(k-1) = \dot{q}_d(k-1) - [\dot{q}(k-1) - \dot{q}(k-2)]/T_s \quad (3.6)$$

$$e(k-1) = q_d(k-1) - q(k-1) \quad (3.7)$$

where q and q_d represent the actual and desired joint angles, k denotes the sample instant, and T_s is the sample period.

The regressor, Y , is based on the structure of the manipulator system dynamics, including reflected actuator inertia and friction terms, and is dependent on only the desired trajectory positions, velocities, and accelerations, not actual trajectory characteristics.

The columns of the regressor matrix are arranged based on a priori knowledge of PUMA dynamics [13]. The first three columns of \mathbf{Y} (and elements of \mathbf{a}) are directly related to gravitational torques. Inertial parameters are arranged according to payload sensitivity analysis. Columns $\mathbf{Y}(11 - 19)$ are viscous, Coulomb, and static friction terms.

The regressor matrix was expanded from a 3×28 matrix listed in [13] to a 3×31 matrix to incorporate the three static friction terms. These terms were inserted as elements $\mathbf{Y}(1, 17)$ through $\mathbf{Y}(3, 19)$ as shown:

$$\begin{bmatrix} e^{-\tau_0|\dot{\theta}_{1d}|}\text{sign}(\dot{\theta}_{1d}) & 0 & 0 \\ 0 & e^{-\tau_0|\dot{\theta}_{2d}|}\text{sign}(\dot{\theta}_{2d}) & 0 \\ 0 & 0 & e^{-\tau_0|\dot{\theta}_{3d}|}\text{sign}(\dot{\theta}_{3d}) \end{bmatrix}$$

The term τ_0 is a lubrication coefficient which is selected to be 40 [5] and $\dot{\theta}_{1d}$ is the desired velocity of the first joint.

The remaining columns of \mathbf{Y} were shifted to be $\mathbf{Y}(i, 20)$ through $\mathbf{Y}(i, 31)$ where i is the joint number. Individual static friction terms are independent of other joint trajectories, but other terms of the regressor are coupled. Differences in one link's performance attributed to this compensation may still indirectly affect the performance and adaptation of other links.

The parameter vector and regressor matrix are assumed to have two components. The "known" parameters, $\hat{\mathbf{a}}_n$, will remain constant during program execution while the unknown parameters, $\hat{\mathbf{a}}$, will be adapted. The elements of the regressor matrix corresponding to the known parameters are denoted by \mathbf{Y}_2 . Feedforward torque components derived from the known parameters and the regressor are precomputed and stored off-line.

The columns of the regressor matrix corresponding to the unknown parameters are denoted by \mathbf{Y}_1 . The controller software is written such that the column size of \mathbf{Y}_1 is selected by the program operator. The unknown parameters are initialized to operator-determined values. This portion of the feedforward torque is calculated on-line [12]. The

total feedforward torque is expressed by:

$$\tau_{ff} = Y_1(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)\hat{a} + Y_2(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)\hat{a}_n \quad (3.8)$$

The unknown parameters are adapted by a Desired Compensation Adaptive Law [19]:

$$\hat{a} = \int \Gamma^{-1} Y^T(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)[(\dot{\theta}_d - \dot{\theta}) + \Lambda(\theta_d - \theta)] \quad (3.9)$$

This adaptation is digitally implemented by[13]:

$$\hat{a}(k) = \int_0^{T_s} \Gamma^{-1} Y_1^T(q_d(k), \dot{q}_d(k), \ddot{q}_d(k))[\dot{e}(k-1) + \Lambda e(k-1)] \quad (3.10)$$

The vector Λ is selected to be the ratio of the position feedback loop gain over the velocity feedback loop gain. Selection of Γ^{-1} is discussed in a later section.

3.4 Experimental Evaluation

3.4.1 Introduction. Initial testing of the AMBC algorithm examined the effects of adding three more friction parameters. These tests were conducted with an initialized parameter vector and with an uninitialized vector. The values used to initialize the parameter vector came from [14]. The parameters were also rearranged to conduct tests using only the three aforementioned gravitational terms, one inertial parameter, and all nine friction terms. Tests were conducted over a variety of trajectories, listed in Table 3.1, under zero payload conditions. Test results are in Section 3.5. Motor saturation constraints limited testing with a 2 kg payload to Trajectories 1 and 3 only.

Trajectories 1, 2, and 3 all experience angular position movement of $(95^\circ, 45^\circ, -105^\circ)$. Trajectory 1 undergoes this movement in 1.5 seconds instead of the 2.0 seconds used in Trajectories 2 and 3 and is the standard evaluation trajectory used in previous PUMA evaluations [13]. The desired trajectory velocity and acceleration components will be identical for Trajectories 2 and 3. These three test trajectories will allow testing to consider the effects of different starting positions and trajectory speeds.

Table 3.1. Test Trajectories (degrees)[13]

Number	Start	Finish	Time (sec)
1	-50,-135,135	45,-90,30	1.5
2	-50,-205,90	45,-160,-15	2.0
3	0,-180,180	95,-135,75	2.0
4	0,-90,90	-95,-135,-15	2.0
5	0,-45,135	-95,-90,20	2.0
6	0,-90,90	95,-135,195	1.5

Trajectories 4 and 5 apply identically generated trajectory commands to different initial conditions. When compared to Trajectories 2 and 3, the initial positions differ and the movement of Joints 1 and 2 is opposite. (Trajectories 4 and 5 move $(-95^\circ, -45^\circ, -105^\circ)$). The respective desired position and acceleration terms should also differ in sign from those of Trajectories 2 and 3. These two test trajectories permit testing to consider the effects of different starting positions and direction of motion.

The movement of Trajectory 6 is similar to Trajectory 1. Joint 3 moves in the opposite direction while Joint 1 and 2 movement is the same. Desired velocity and acceleration terms for Joint 3 should have opposite signs. This trajectory can be used to determine the effects of different starting positions, direction of motion, and trajectory speed.

3.4.2 Sixteen-Parameter Adaptation Testing. All trajectories in Table 3.1 were run to emulate Leahy and Whalen's adaptive algorithm using 16-parameters in the $\hat{\mathbf{a}}$ vector [13], validate the modified software, and account for any minor changes in the manipulator since their study. This was accomplished by setting the first three terms of $\hat{\mathbf{a}}_n$, which correspond to the static friction terms, to zero. Trajectories 1 and 3 were also run carrying a 2 kg payload. The diagonal Γ^{-1} matrix was set to Leahy and Whalen's values of (120,120,120,0,90,90,90,15,150,5,80,30,15,80,80,80). The zero in the fourth element denotes that the fourth parameter of $\hat{\mathbf{a}}$ will not be adapted, but remain at its initial value. The effects on adaptation caused by knowledge (or lack thereof) of the values of $\hat{\mathbf{a}}$ were explored by using initial approximations of $\hat{\mathbf{a}}$ and \mathbf{a}_n equal to a set of previously determined values (initialized) or $\hat{\mathbf{a}} = \hat{\mathbf{a}}_n = \mathbf{0}$ (uninitialized).

3.4.3 Nineteen-Parameter Adaptation Testing.

3.4.3.1 *Nineteen-Parameter Tuning.* In this series of evaluations, the AMBC algorithm was used in conjunction with a 19 parameter \hat{a} vector. The static friction terms were now part of the unknown parameter vector. An a priori estimate of $\hat{a}(17-19) = 1$ was used in determining the best set of $\Gamma^{-1}(17-19)$ for the adaptation expressed in Equation (3.9). The first 16 values of $\Gamma^{-1}(1-16)$ were left the same as noted in the last section.

Trajectory 1 was run several times using different values for $\Gamma^{-1}(17)$. The values of $\Gamma^{-1}(18-19)$ were set to zero. The value that produced the best performance was selected, then several more runs were executed to determine $\Gamma^{-1}(18)$, and finally $\Gamma^{-1}(19)$. The tuning procedure was repeated for Trajectory 1 with a 2 kg payload. The values of $\Gamma^{-1}(17-19)$ were chosen to be (200,15,15).

3.4.3.2 *Nineteen-Parameter Adaptation Runs.* All of the trajectories listed in Table 3.1 were run using the Γ^{-1} matrix obtained from tuning the AMBC algorithm for 19 parameters. The test conditions included running the AMBC algorithm with and without payloads and:

1. with \hat{a} parameters initialized (set to predetermined values). Then:
 - (a) Evaluate the algorithm for single-run performance.
 - (b) Conduct multiple runs with \hat{a} set to values derived during the previous runs to demonstrate the value of learning in a repetitive task environment.
2. with \hat{a} parameters uninitialized (set to zero). Then:
 - (a) Evaluate the algorithm for single-run performance.
 - (b) Conduct multiple learning runs to learn the \hat{a} parameters.

3.4.4 *Thirteen-Parameter Adaptation Testing.* The last set of tests undertaken using the AMBC algorithm necessitated rewriting the test software. These tests evaluated the effects of using just three gravity parameters, one inertial parameter, and all nine friction

parameters as the first thirteen elements of $\hat{\mathbf{a}}$. The regressor matrix was again reconfigured to place the friction compensation terms originally in $\mathbf{Y}(i, 11 - 19)$ into $\mathbf{Y}(i, 5 - 13)$, respectively, where i is the joint number. Regressor components $\mathbf{Y}(i, 5 - 10)$ were replaced as $\mathbf{Y}(i, 14 - 19)$. The remaining regressor terms were left in their initial positions. The regressor and manipulator parameter terms were moved so that all of the terms now associated with the known parameters could be set to their predetermined values and the corresponding feedforward torques precalculated and stored.

3.4.4.1 Thirteen-Parameter Tuning. The AMBC algorithm was now tested with a 13-parameter $\hat{\mathbf{a}}$ vector. The Γ^{-1} parameters pertaining to the gravity and inertial terms were left at (120,120,120,0). These values were used by Leahy and Whalen in previous tests with 7, 10, and 13 element unknown parameter vectors [12, 13]. As in 19-parameter tuning, an estimate of $\hat{\mathbf{a}}(11 - 13) = 1$ was used for the static friction terms. Trajectory 1 was run several times using different values for $\Gamma^{-1}(5 - 7)$. The values of $\Gamma^{-1}(8 - 13)$ were set to zero. The values of $\Gamma^{-1}(5 - 7)$ that produced the best performance were selected, then several more runs were performed to determine $\Gamma^{-1}(8 - 10)$, and finally $\Gamma^{-1}(11 - 13)$. The values of $\Gamma^{-1}(5 - 13)$ were chosen to be (600,600,200,600,600,200,200,200,200).

3.4.4.2 Thirteen-Parameter Adaptation Runs. All of the trajectories listed in Table 3.1 and the two payload bearing trajectories mentioned in Section 3.4.1 were run using the Γ^{-1} matrix obtained from tuning the AMBC algorithm for 13 parameters. The test conditions were the same as in the 19-parameter case. The next section discusses the results obtained from the 16-, 19-, and 13-parameter tests.

3.5 Adaptive Model-Based Control Algorithm Test Results

3.5.1 Introduction. All test trajectories given in Table 3.1 were run under no-load conditions. Trajectories 1 and 3 were also run carrying a 2 kg payload. All test trajectories were run several times using the adaptive parameters derived from the previous learning run as initial parameter estimates.

This discussion compares tests completed under similar test scenarios. The 13-, 16-, and 19-parameter adaptive algorithms were run under the conditions of:

- uninitialized parameters and no load
- uninitialized parameters and a 2 kg load
- initialized parameters and no load
- initialized parameters and a 2 kg load

The adaptive algorithm test results were also compared to evaluations that used a Single Model-Based Controller.

The Single Model-Based Controller (SMBC) for these comparisons was formed from the model-based controller used in the AMBC algorithm. The SMBC controller used the same PD feedback torques as the AMBC controllers and feedforward torques based solely on the desired trajectory attributes and assumed manipulator parameters as shown in Equations (2.3) and (2.20). The entire manipulator parameter vector \hat{a} was initialized to predetermined values and the Γ^{-1} vector was set to zero to prohibit any of the parameters from being adapted.

3.5.2 Single Run Tests With Uninitialized Parameters. Figures 3.1-3.3 are representative of the plots in Appendix D. These figures show that the uninitialized 13-parameter AMBC algorithm, of which nine parameters are friction parameters, produces better trajectory responses than the uninitialized 16- or 19-parameter AMBC controllers for single run tests. This may be attributed to the large gains in the 13-parameter Γ^{-1} vector. The manipulator parameters may adapt more quickly than those determined using the 16- or 19-parameter algorithms' Γ^{-1} vectors. The 16- and 19-parameter error plots trace each other closely throughout the entire set of single run tests with uninitialized parameters. The addition of the three static friction terms to the 16-parameter AMBC controller does not have a significant influence on trajectory tracking for single run uninitialized tests.

3.5.3 Learning Runs With Uninitialized Parameters. The 13-, 16-, and 19-parameter AMBC controllers were run several times to allow the algorithms to learn the manipulator

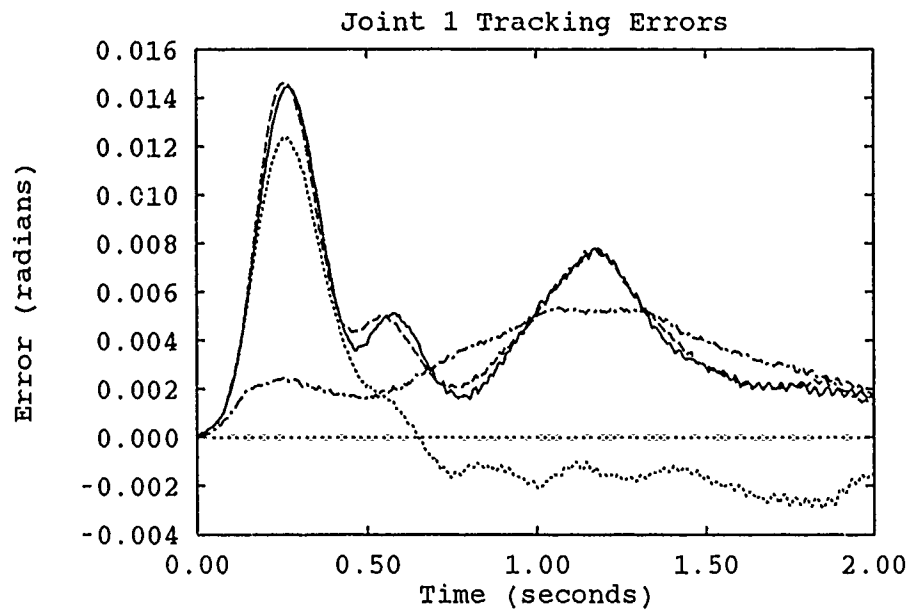


Figure 3.1. Comparison of Uninitialized Single Run Tests - Trajectory 3

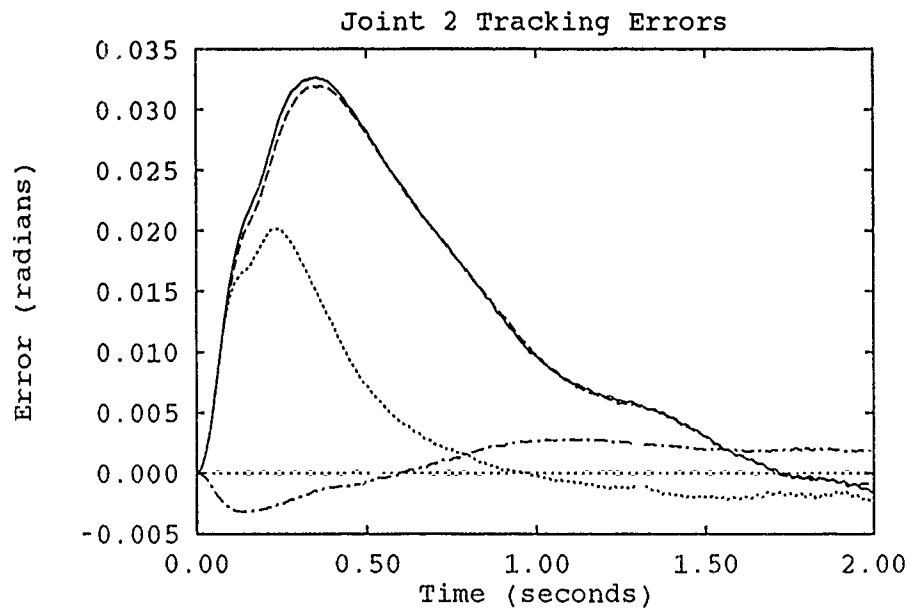


Figure 3.2. Comparison of Uninitialized Single Run Tests - Trajectory 3

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

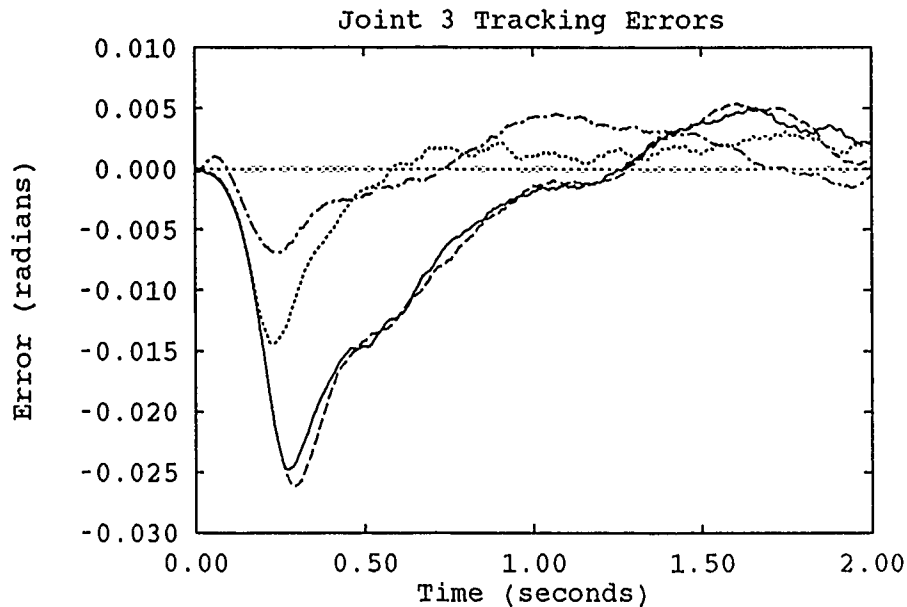


Figure 3.3. Comparison of Uninitialized Single Run Tests - Trajectory 3

—	19 Parameters	13 Parameters
- - -	16 Parameters	- · - · -	SMBC

parameters. Appendices A, B, and C contain plots of the tracking errors developed in each of the trajectories as the parameters were being learned. Appendix E displays comparisons of the final results of learning for all test trajectories and provides a comparison to SMBC testing. Appendices B and C plots show that the overall tracking errors are reduced and peak errors are reduced by as much as 80% (see Figures B.10 and C.10) of the original errors. Figures 3.4-3.6 highlight the results of learning with uninitialized parameters for Trajectory 4. The uninitialized 19- and 16-parameter AMBC controllers produce results superior to the SMBC and 13-parameter algorithms. In most test cases the 19-parameter AMBC controller is closely tracked by the 16-parameter controller. In the remainder of the tests the 19-parameter controller is better. The extra friction terms slightly improve the trajectory response.

Even after learning, the uninitialized 13-parameter AMBC controller is inferior to the SMBC controller. Over half of the plots in Appendix A show that the learning runs actually caused this controller to produce worse tracking errors than those experienced in single run

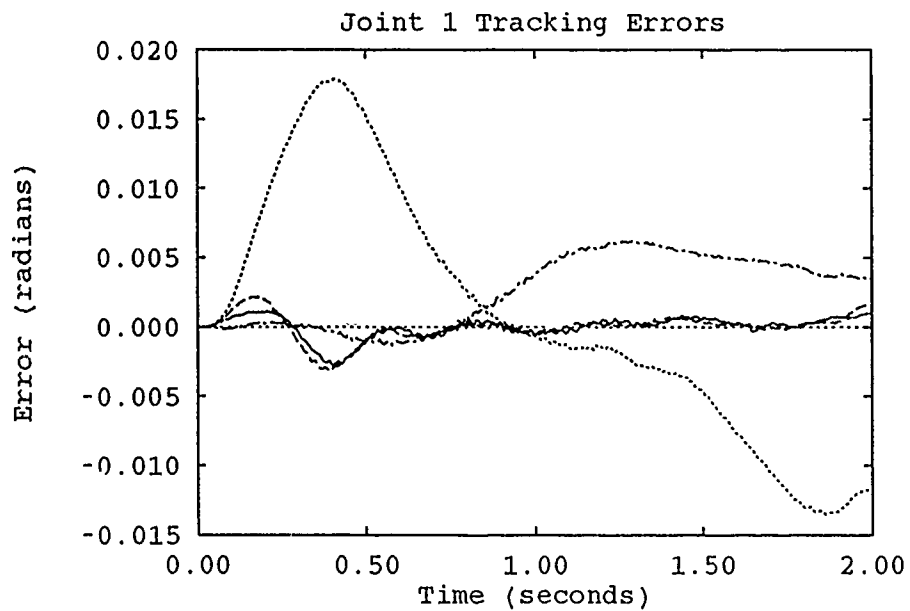


Figure 3.4. Comparison of Uninitialized Runs After Learning - Trajectory 2

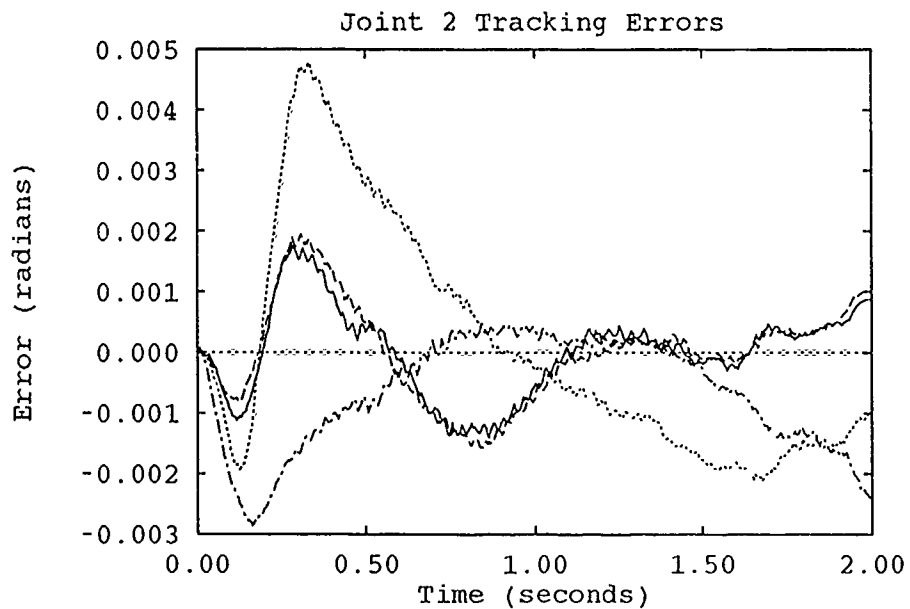


Figure 3.5. Comparison of Uninitialized Runs After Learning - Trajectory 2

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - . -	SMBC

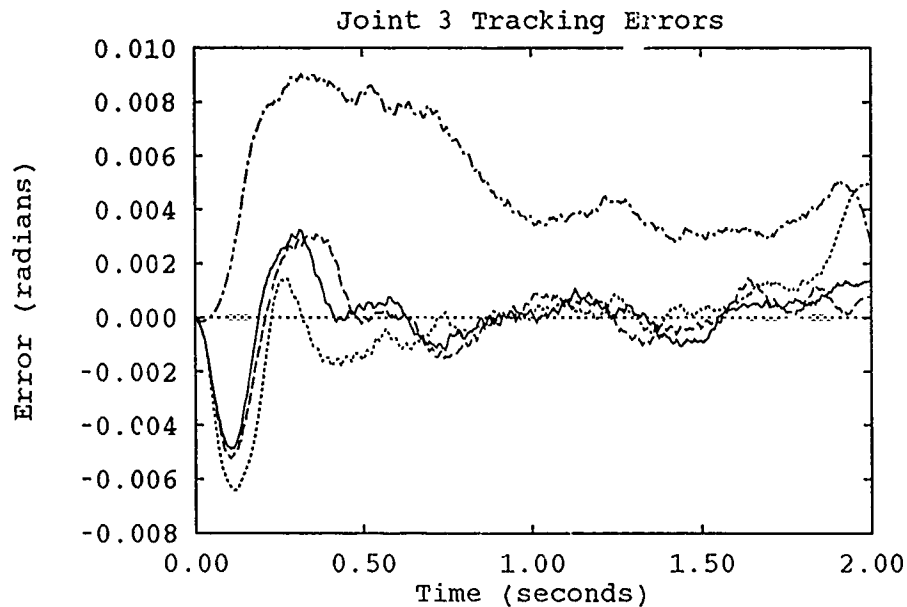


Figure 3.6. Comparison of Uninitialized Runs After Learning - Trajectory 2

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

tests. The 13-parameter AMBC may have made an initial step in the wrong direction when trying to estimate its viscous friction parameters and forced to stay in that area because of the heavy emphasis on friction terms imposed by the corresponding large Γ^{-1} values. The results of 13-parameter adaptation with initialized parameters are much improved and the manipulator parameter adaptations are more in line with the initialized values. The viscous friction terms are either positive or much closer to zero than the estimates made in the uninitialized cases. Retuning the first four Γ^{-1} parameters before tuning the friction components may have relieved some of this error, and tuning during learning runs instead of single trajectory testing could help improve this algorithm's performance.

Appendices L through N contain tables listing the parameters learned through repetitive testing. Appendix O provides a side-by-side comparison of comparable parameters. Table 3.2 shows the results of no-load Trajectory 1 testing. The three columns on the right contain the manipulator parameters determined after the ninth learning runs of the uninitialized parameter tests. These algorithms should be trying to find the same \hat{a} values.

Table 3.2. Learned Parameter Values - Trajectory 1

Number	Nominal	Trajectory 1 Initialized			Trajectory 1 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.30	0.85	1.04	0.87	-4.96	5.81	6.78
2	-52.09	-54.86	-53.83	-53.15	-30.73	-17.64	-17.88
3	-7.53	-7.30	-8.81	-9.63	-10.12	-8.46	-8.88
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.03		0.23	0.38		-1.08	-1.18
6	-1.57		-2.86	-2.77		-3.34	-3.36
7	2.10		2.95	3.40		0.65	0.82
8	-0.03		-0.70	-0.65		-1.20	-1.24
9	0.67		2.99	3.04		11.41	11.58
10	-0.12		-0.23	-0.16		-1.90	-1.80
11	4.50	-1.73	4.81	4.23	-36.53	7.04	7.00
12	3.50	-0.23	4.08	4.16	-23.76	3.58	3.61
13	3.50	1.09	3.45	3.52	-9.25	3.18	3.33
14	5.95	7.62	7.46	7.35	10.42	5.02	4.74
15	6.82	10.58	9.51	9.88	20.56	17.00	17.13
16	3.91	4.30	5.86	6.02	4.72	10.14	9.45
17	1.00	0.21		1.64	-5.11		1.22
18	1.00	-0.34		0.87	-2.43		-0.16
19	1.00	0.37		0.91	-5.90		-0.33

The manipulator parameters found by the 16- and 19-parameter AMBC controllers are similar from top to bottom. There is no correlation between these manipulator parameters and those developed by the 13-parameter AMBC controller. The discrepancies of the 13-parameter controller may have been caused by using fewer parameters to produce the same level of robot control and needing different values to attain the goal, or by the large Γ^{-1} vector causing the manipulator parameters values to change much more drastically than in the 16- or 19-parameter cases. The 13-parameter algorithm suffers severely when half of its parameters are given an incorrect sign as a result of adaptation. Other tables in Appendix O show that the uninitialized 19- and 16-parameter controllers produce results similar to each other for each trajectory. The parameter values change from trajectory to trajectory which is consistent with the findings of Leahy and Whalen [13].

3.5.4 Single Run Tests With Initialized Parameters. Trajectory errors resulting from testing with initialized parameters are at least 50% smaller than those experienced running single run tests with uninitialized parameters. Figures 3.7-3.9 are representative of the no-load trajectories in Appendix I. For every no-load trajectory test run, the 13-parameter AMBC algorithm produced the best response, except Trajectory 6 - Joint 3 where large oscillations developed in the middle of the trajectory tracking error. The 16- and 19-parameter models develop tracking responses that are similar to each other. As expected, the SMBC controller was ineffective when compared to any of these AMBC controllers.

The 2 kg payload test runs all experienced oscillations and degradation over the no-load tests. The 16-parameter algorithm developed the most favorable set of error profiles. The 19-parameter system was a very close second and the 13-parameter model third. The 13-parameter algorithm may have been penalized for its smaller set of adapting parameters. The single run tests utilizing initialized parameters were nearly comparable to the uninitialized parameter cases after learning. These sets of initialized runs show that there is no substitute for good initial estimates.

3.5.5 Learning Runs With Initialized Parameters. AMBC controllers with 13, 16, and 19 initialized parameters were allowed to use the parameters learned during one run as the initial parameter estimate for the next run. Appendices F, G, and H contain plots of the errors generated during the learning process. As was the case with uninitialized parameters, learning runs for the initialized 13-parameter AMBC controller did not necessarily mean that the trajectory tracking response was going to improve. Several of the figures in Appendix F show that the errors increased as the controller was supposed to be learning the manipulator parameters. Neither the 16- nor 19-parameter AMBC controller could improve tracking response for Joint 3 of Trajectory 6. These controllers did improve or maintain the tracking responses for all of the other test cases. Some of the tracking error plots in Appendices G and H showed peak error reductions of more than 60%.

The learning process was complete by the fifth run for all three algorithms. Figures 3.10-3.12 show one set of plots contained in Appendix J. The appearance of oscillations in

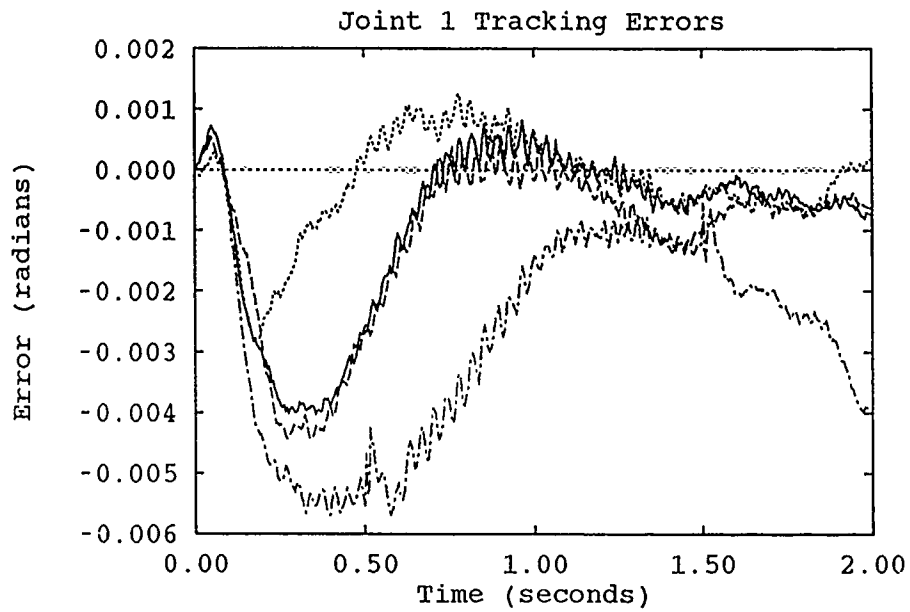


Figure 3.7. Comparison of Initialized Single Run Testing - Trajectory 4

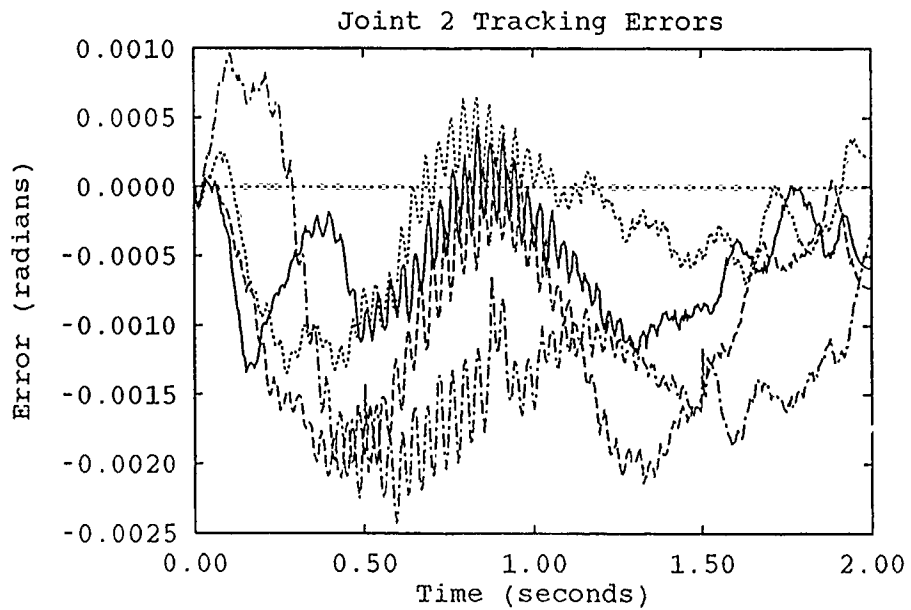


Figure 3.8. Comparison of Initialized Single Run Testing - Trajectory 4

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

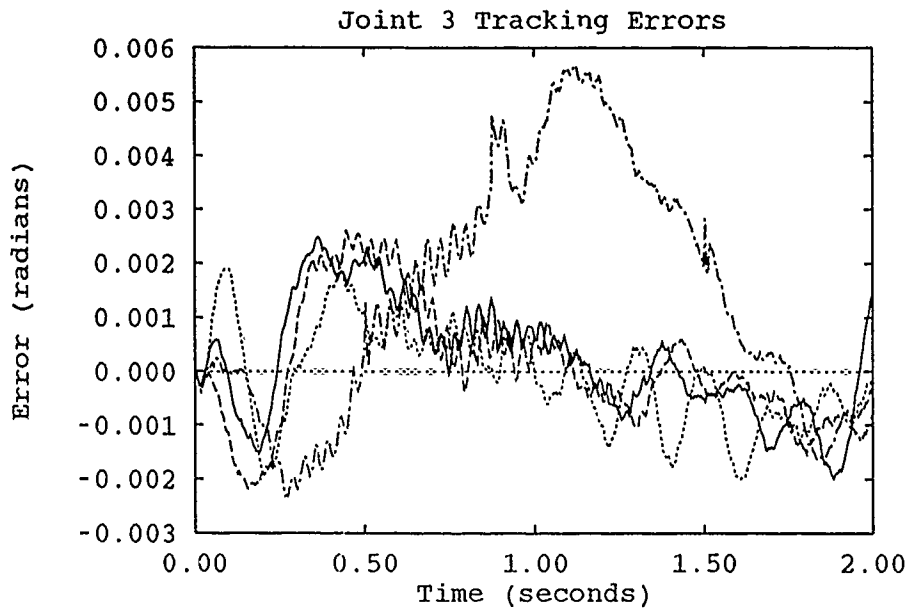


Figure 3.9. Comparison of Initialized Single Run Testing - Trajectory 4

—	19 Parameters	13 Parameters
---	16 Parameters	-.-.-	SMBC

some plots is due to the small range covered by the vertical axis of the plots. Figure 3.11 covers an entire range of .003 radians from top to bottom.

The 16- and 19-parameter algorithms usually trace similar error profiles. The initialized 13-parameter AMBC controller is surprisingly capable. It provided the best control for all of the joints of the robot for Trajectory 3. This controller, however, was the worst controller for the payload bearing and high speed trajectories. The 19-parameter controller worked best, and outperformed the 16-parameter controller, on the heavier Links 1 and 2. The 16-parameter controller was best utilized for Joint 3.

The three columns in the center of Table 3.2 contain the results of no-load Trajectory 1 testing for learning with initialized parameters. Again the manipulator parameters found by the 16- and 19-parameter AMBC controllers are similar. There are sign changes in some of the parameters developed by the 13-parameter AMBC controller. It has decided that the viscous friction terms for Joints 1 and 2 should be negative. Tables O.4, O.5, and O.8 show the 16- and 19-parameter controllers determining that parameter#1 as a negative number.

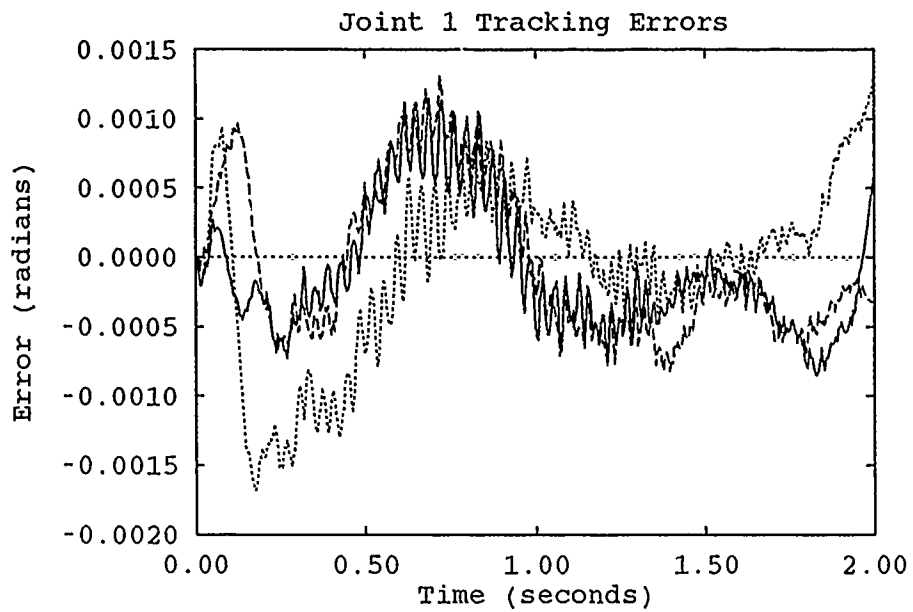


Figure 3.10. Comparison of Initialized Runs After Learning - Trajectory 4

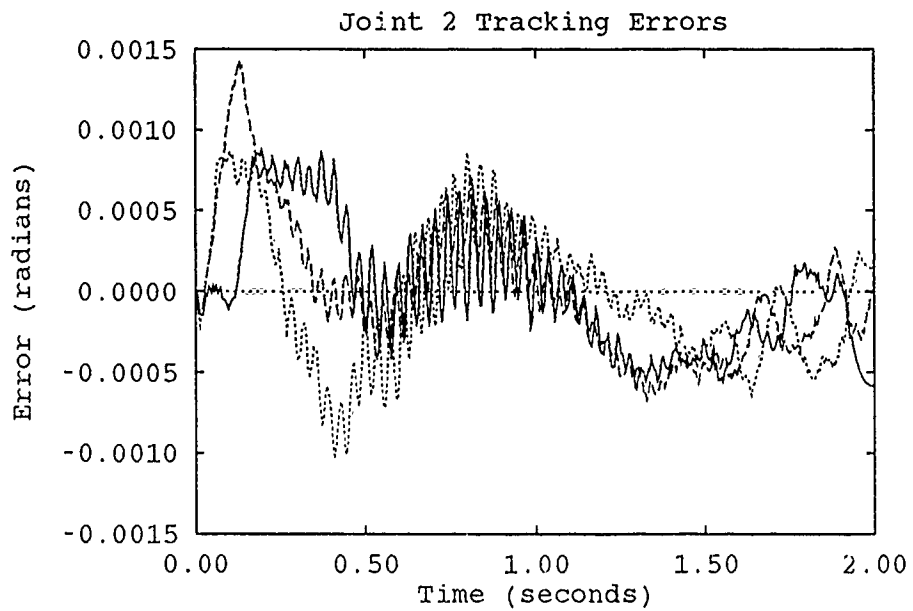


Figure 3.11. Comparison of Initialized Runs After Learning - Trajectory 4

—	19 Parameters	13 Parameters
- - -	16 Parameters		

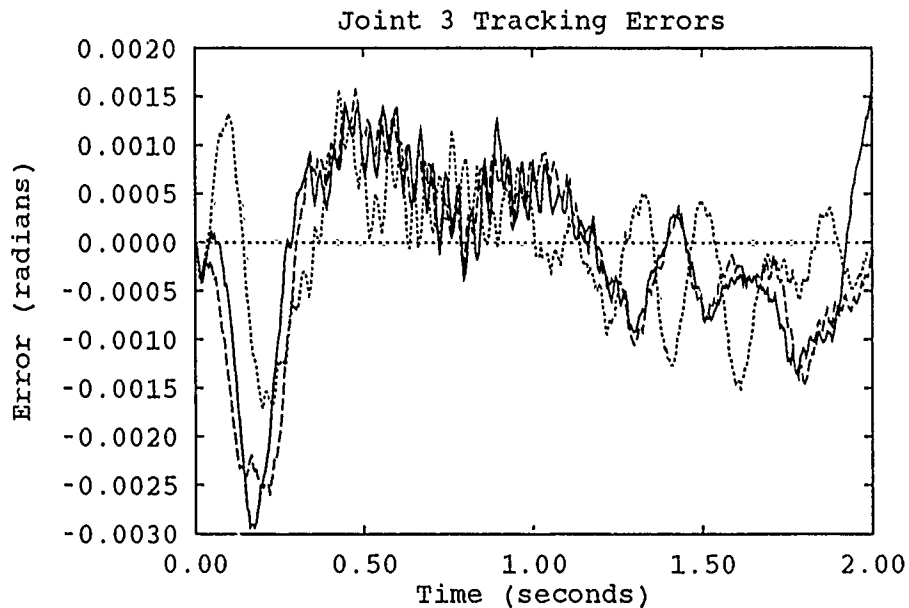


Figure 3.12. Comparison of Initialized Runs After Learning - Trajectory 4

—	19 Parameters	13 Parameters
- - -	16 Parameters		

The nominal values were positive. Similar changes in the signs of several parameters are probably responsible for the reduction in the response for the 13-parameter AMBC.

3.5.6 Comparison of Initialized and Uninitialized Learning Runs. Figures 3.13-3.15 from Appendix K illustrate the value of initialized parameters over uninitialized parameters for 16- and 19-parameter controllers. The 13-parameter controller is inferior to these two because of its shortcomings for the high-speed and loaded tests. These figures show that the tracking errors can be reduced to approximately .001 radian. If computational loading is an issue, the 16-parameter controller performs nearly as well as the 19-parameter controller.

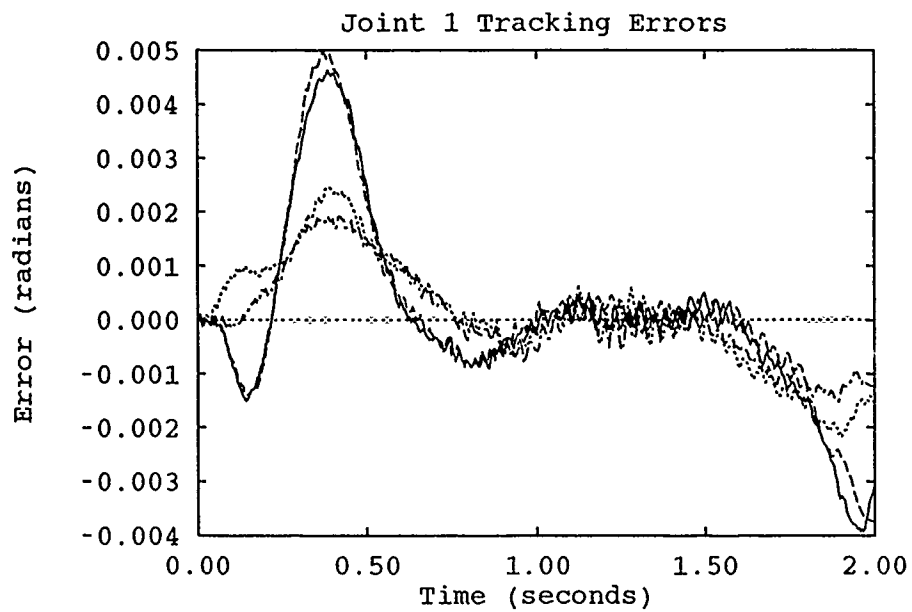


Figure 3.13. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5

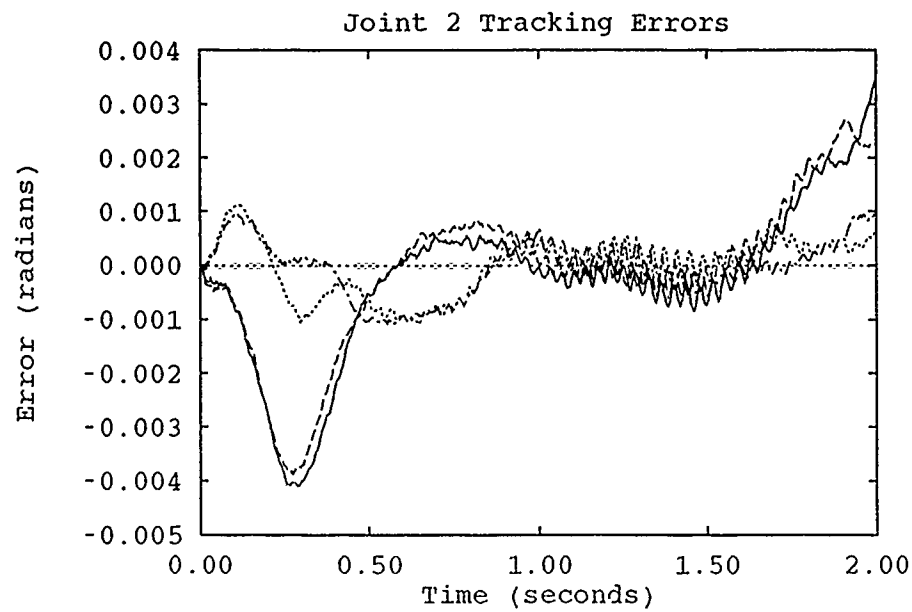


Figure 3.14. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5

—	19 Uninitialized Parameters	19 Initialized Parameters
----	16 Uninitialized Parameters	-.-.-.-	16 Initialized Parameters

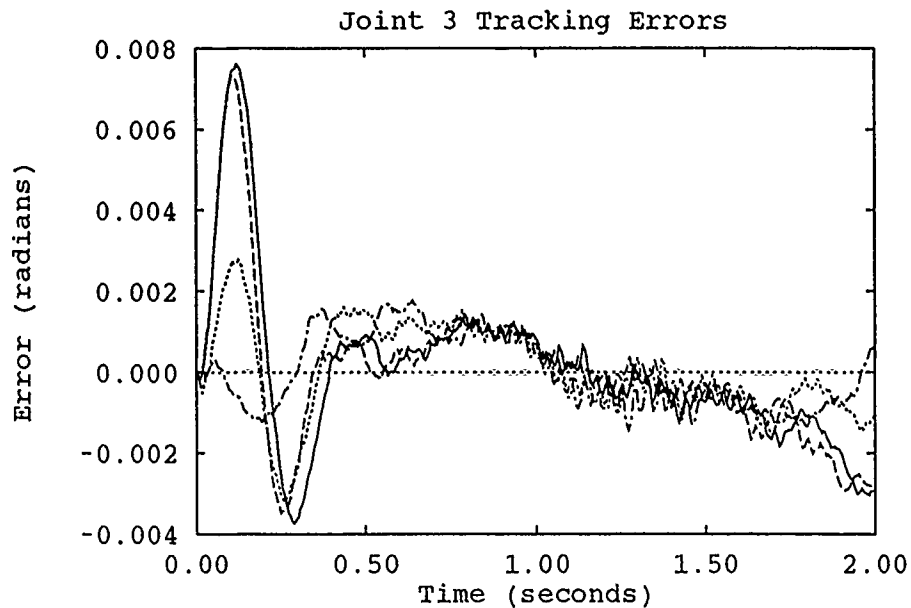


Figure 3.15. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5

—	19 Uninitialized Parameters	19 Initialized Parameters
---	16 Uninitialized Parameters	-.-.-	16 Initialized Parameters

3.6 Summary

This chapter discussed the evaluation of an Adaptive Model-Based Controller. Tests were conducted over a variety of trajectories with different numbers of initialized and uninitialized parameters. Uninitialized tests evaluated the algorithms' ability to develop manipulator parameters in the absence of any prior information about the parameters. The addition of extra friction parameters did not improve the response of the basic 16-parameter AMBC during single run testing. Learning tests did reduce the peak errors of the single-run tests by as much as 80%.

Initialized tests provided the controller with an a priori estimate of manipulator parameters. The controllers were also able to compute an additional feedforward torque component based on known non-adapting parameters. The parameter knowledge helped reduce tracking errors by 50% with respect to comparable uninitialized tests. The 13-parameter AMBC emphasizes friction components. It produced the best single-run responses, but was penalized for its smaller parameter size and nearly diagonal structure of the regres-

sor matrix in the learning tests. The 16- and 19-parameter AMBCs reduced most peak tracking errors to less than one milliradian. There was no decided advantage to using 19 parameters over 16 for initialized testing, and therefore no need to implement the static friction parameters.

IV. Evaluation of Decentralized Digital Control Algorithms

4.1 Introduction to the Discussion of Tarokh's Algorithm

In a pair of publications, Tarokh has presented the development of a Popov hyperstability-based discrete-time adaptive control scheme [22, 23]. These papers were discussed in Chapter 2. His initial evaluations were conducted over trajectories that moved Joints 2 and 3 of a PUMA-560 manipulator from $(0^\circ, 0^\circ)$ to $(50^\circ, -50^\circ)$. The time of the trajectory was 2 seconds. Attempts were made to validate those results. Restrictions imposed on our Puma led to our test trajectories starting at $(-180^\circ, 180^\circ)$ and finishing at $(-130^\circ, 130^\circ)$. The design parameters Tarokh selected are listed in Table 4.1.

Table 4.1. Tarokh's Design Parameters

Feedback terms:	$E_{1P} = 0$ $E_{1I} = 2I_{2 \times 2}$	$E_{2P} = 0$ $E_{2I} = 2I_{2 \times 2}$
Feedforward terms:	$F_{0P} = 0$ $F_{1P} = 0$ $F_{2P} = 0$	$F_{0I} = 0$ $F_{1I} = 0$ $F_{2I} = 0$
Constants:	$\alpha_2 = 200$	$\alpha_3 = 100$
Eigenvalues:	$\lambda_1 = -0.5I_{2 \times 2}$	$\lambda_2 = -0.5I_{2 \times 2}$
Auxiliary terms:	$\beta_P = 50I_{2 \times 2}$	$\beta_I = 0.5I_{2 \times 2}$
Initial Conditions:	$\eta_2(0) = -25$	$\eta_3(0) = 3$

The subscripts for the η and α terms refer to the joint number. The subscripts on the E and F matrices denote which P or Q matrix of Equation (2.42) they are used to create. The implementation of Equation (2.42) is discussed in the next section. The values of the constant α terms were not given in the section of Tarokh's paper covering experimental results [22]. The α values listed in Table 4.1 are taken from his computer simulation section. All values in this table are to be specified by the designer to meet the needs of the particular manipulator. In the attempt to reproduce Tarokh's plots, the α values used were (100,50) and the λ values were changed to -0.8. The initial conditions listed in Table 4.1 were used. If this control algorithm is not dependent on its trajectory, initial conditions should apply to any trajectory selected. Feedforward compensation terms were not listed

in either the computer simulation or experimental results section and were therefore taken to be zero.

The evaluation runs were accomplished under the same conditions for adjacent link movement and payload parameters as Tarokh used. Figure 4.1 shows the desired and actual trajectories covered by the Joint 2 of the manipulator under the conditions of no movement in Joint 3, with and without payload. Figure 4.2 shows the desired and actual Joint 2 trajectories with and without Joint 3 movement under no-load condition. The plots of actual trajectory errors shown in Figures 4.3 and 4.4 are more revealing. These plots show how large the tracking errors produced really are.

Although the trajectories generated by our manipulator do not demonstrate the same smoothness shown by Tarokh's experiments, Figures 8 and 9 of [22], they supplied enough incentive to explore the algorithm further in the hope of attaining suitable parameters to provide accurate trajectory tracking. The next section describes the approach taken in tuning and testing Tarokh's algorithm.

4.2 Examination of Tarokh's Algorithm

The discrete-time adaptive control algorithm implemented in this effort is given in Equation (2.42) as:

$$\begin{aligned} u(k) = & P_1(k)\theta_e(k-1) + P_2(k)\theta_e(k-2) + Q_0(k)\theta_d(k) + Q_1(k)\theta_d(k-1) \\ & + Q_2(k)\theta_d(k-2) + \eta(k) \end{aligned} \quad (4.1)$$

The control algorithm can be divided into feedback, feedforward, and auxiliary torque components as follows:

$$\text{Feedback Torque:} \quad P_1(k)\theta_e(k-1) + P_2(k)\theta_e(k-2)$$

$$\text{Feedforward Torque:} \quad Q_0(k)\theta_d(k) + Q_1(k)\theta_d(k-1) + Q_2(k)\theta_d(k-2)$$

$$\text{Auxiliary Torque:} \quad \eta(k)$$

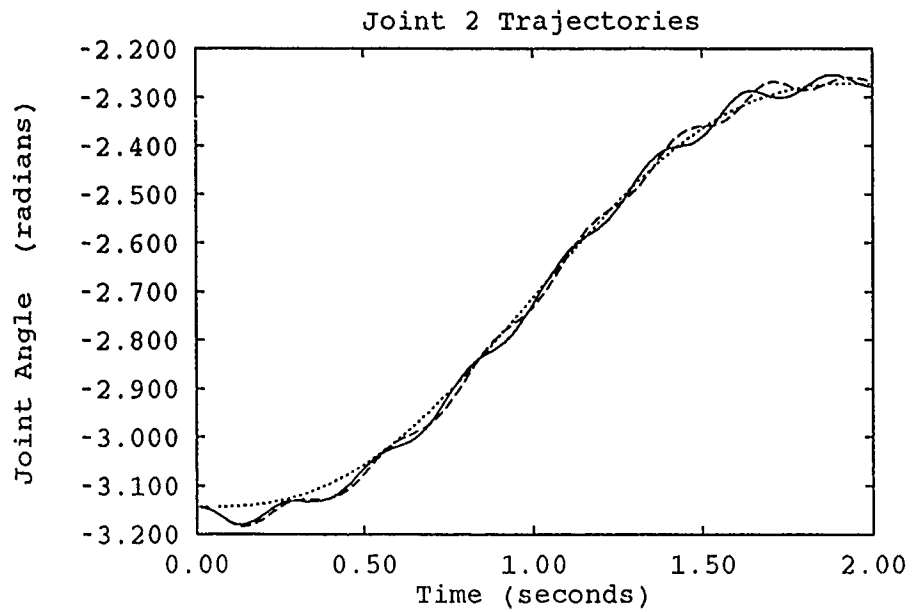


Figure 4.1. Joint 2 Trajectories with no Joint 3 Movement

—	No load trajectory	Desired trajectory
----	2 kg load trajectory		

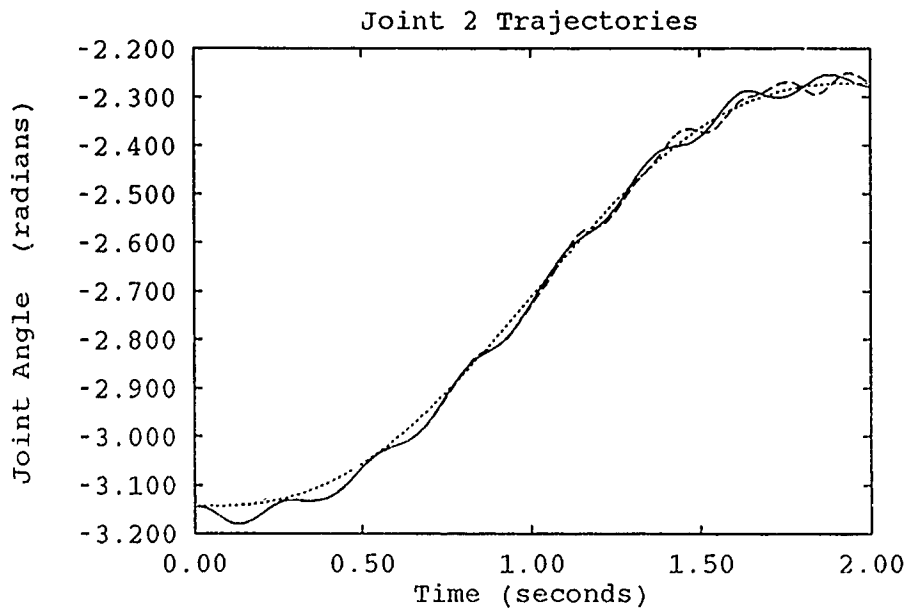


Figure 4.2. Joint 2 Trajectories with no Payload

—	No Joint 3 movement	Desired trajectory
----	With Joint 3 movement		

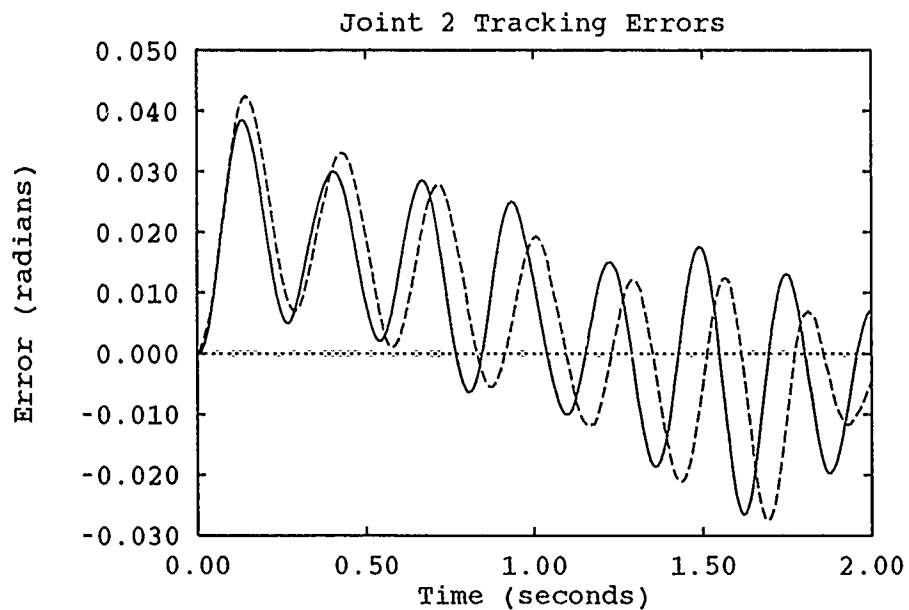


Figure 4.3. Joint 2 Errors with no Joint 3 Movement

—	No load trajectory
- - -	2 kg load trajectory

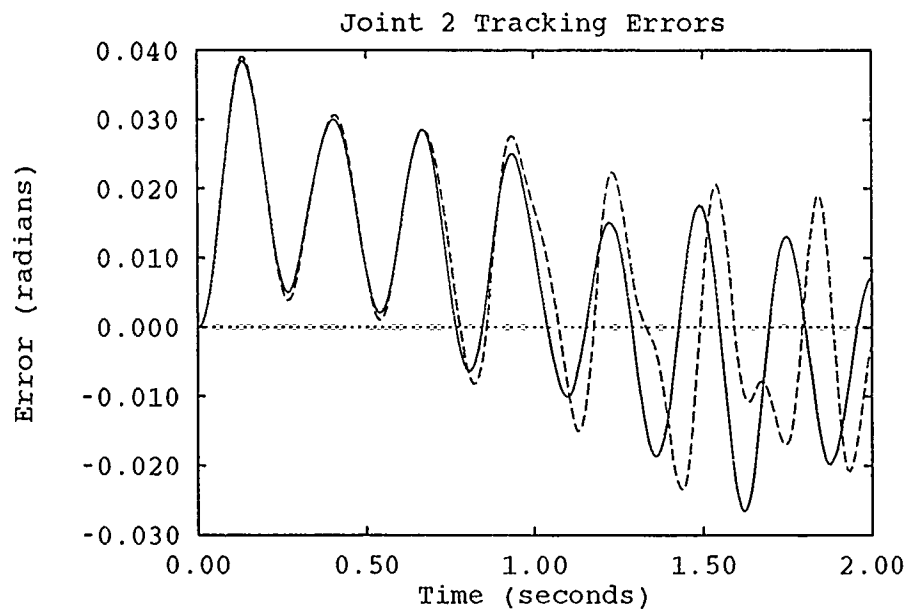


Figure 4.4. Joint 2 Errors with no Payload

—	No Joint 3 movement
- - -	With Joint 3 movement

where the components are modeled by:

$$\begin{aligned} \mathbf{P}_x(k) &= \mathbf{P}_x(k-1) + \hat{\boldsymbol{\theta}}_e(k) \boldsymbol{\theta}_e(k-x) \mathbf{E}_{xP} \\ &\quad + \hat{\boldsymbol{\theta}}_e(k-1) \boldsymbol{\theta}_e(k-x-1) [\mathbf{E}_{xI} - \mathbf{E}_{xP}] \quad x = 1, 2 \end{aligned} \quad (4.2)$$

$$\begin{aligned} \mathbf{Q}_y(k) &= \mathbf{Q}_y(k-1) + \hat{\boldsymbol{\theta}}_e(k) \boldsymbol{\theta}_d(k-y) \mathbf{F}_{yP} \\ &\quad + \hat{\boldsymbol{\theta}}_e(k-1) \boldsymbol{\theta}_d(k-y-1) [\mathbf{F}_{yI} - \mathbf{F}_{yP}] \quad y = 0, 1, 2 \end{aligned} \quad (4.3)$$

$$\boldsymbol{\eta}(k) = \boldsymbol{\eta}(k-1) + \beta_P \hat{\boldsymbol{\theta}}_e(k) + (\beta_I - \beta_P) \hat{\boldsymbol{\theta}}_e(k-1) \quad (4.4)$$

The weighted joint angle error $\hat{\boldsymbol{\theta}}_e$ is formed from the matrix parameters $\boldsymbol{\alpha}$, λ_1 , and λ_2 by:

$$\mathbf{R}_2 = \boldsymbol{\alpha} \lambda_1 \lambda_2 (\lambda_1 + \lambda_2) \quad (4.5)$$

$$\mathbf{R}_3 = \boldsymbol{\alpha} (\mathbf{I}_{3 \times 3} + \lambda_1 \lambda_2) \quad (4.6)$$

$$\hat{\boldsymbol{\theta}}_e(k) = \mathbf{R}_2 \boldsymbol{\theta}_e(k-1) + \mathbf{R}_3 \boldsymbol{\theta}_e(k) \quad (4.7)$$

The terms $\boldsymbol{\alpha}$, λ , \mathbf{R}_2 , and \mathbf{R}_3 are constant diagonal matrices and the weighted errors are common to all torque components.

The torque components were ranked in descending order of magnitude, assuming the presence of small joint angle errors, to determine where to start the process of developing parameter values for our particular controller. The order and reasoning are:

1. Auxiliary torque - This term is the weighted sum of the two most recent weighted errors and is only first order as far as the error terms are concerned.
2. Feedforward torques - The three feedforward torque components are products of first order error terms and second order angular position terms. Squaring these quantities eliminates dependence on the sign of the position terms. The position errors are assumed to be small while the position terms can be as large as $\pm\pi$. This term could cause problems if its components grow too large.
3. Feedback torques - These two torques are third order products of the position errors. If errors are in the range of 0.005 radians, these products are on the order of 10^{-7} .

For the desired small errors, the parameters associated with these terms may need to be extremely large to produce any noticeable torques.

4.3 The Tuning Process

Tarokh stated that this algorithm includes several parameters to be chosen by the designer. In this case there are 15 design parameters and 6 initial conditions to be selected for each of three links. In view of the restrictions imposed by a decentralized controller, all design parameters are viewed as vectors or diagonal matrices. This algorithm is not based on the manipulator's dynamics model, as are adaptive model-based algorithms. The parameters are selected solely on the basis of trial and error.

4.3.1 Weighted Error Terms. The first parameters selected for each link were λ_1 and λ_2 over a 2-second trajectory of $(-50^\circ, -135^\circ, 135^\circ)$ to $(45^\circ, -90^\circ, 90^\circ)$. This trajectory is denoted as Trajectory 0. The first runs were conducted with the feedforward component of a Single Model-Based Controller (SMBC) added to limit the errors experienced during the heuristic process of determining an initial set of λ parameters. The α and β terms used in verifying the algorithm were chosen until better values could be identified. Values of λ_1 and λ_2 were tested for each joint in the manner of:

$$(\lambda_1, \lambda_2) = \pm(0.2i + 0.1, 0.2j + 0.1) \quad i = 0, 1, 2, 3, 4 \quad j = 0, 1, 2, 3, 4$$

to test values all over the range of $|\lambda| < 1$. Points of (λ_1, λ_2) in the fourth quadrant form the same weighted errors as the corresponding points in the second quadrant. Therefore, the fourth quadrant was not investigated. The third quadrant (both λ_1 and $\lambda_2 < 0$) provided the best results. Small magnitudes of λ produced oscillations in the manipulator trajectory while larger magnitudes resulted in larger errors over smoother trajectories. Values of λ that produced a compromise of these opposing conditions were selected. This hinted at the need for readjusting λ values as parameters were added.

The constant α terms were recognized as being only multipliers that increase the magnitude of the R_2 and R_3 terms used in calculating the weighted errors. A check of Equations (4.2-4.4) reveals that each P , Q , and η term is a first order function with respect

to $\hat{\theta}_e(k)$ or $\hat{\theta}_e(k-1)$. All α values were set to 1. The constants for β , E, and F terms will all be larger (α times larger) to compensate for this.

4.3.2 Auxiliary Torque Terms. The auxiliary torque component was identified as the largest contributor to total torque, therefore, the β_P and β_I terms were selected first. These terms seemed to influence opposite ends of the trajectory for their particular link. In the case of Links 1 and 2, increasing β_P had the effect of reducing the initial peak errors while inducing oscillations in the final part of the trajectory. Increasing the integral term, β_I , quelled these oscillations but reinforced the initial peak errors. Link 3 demonstrated the opposite behaviors for these terms. These tradeoffs were considered in selecting β_P and β_I .

Initial conditions for the auxiliary input were introduced to examine their effect. The SMBC feedforward term was removed during the β tuning procedure as soon as a set of parameters produced a reasonable response. Figures 4.5 - 4.7 show the trajectory errors incurred during the steps of this tuning process. The oscillatory nature of the tracking error plots is due to the lack of velocity gains in Tarokh's algorithm. Without velocity gain, the algorithm is suitable for regulation, not tracking [10]. After the best set of β terms was selected, the next step was to go back and adjust the λ terms for smoother tracking and then recheck the β terms. The parameters selected during each step of the tuning process are listed in Table 4.2.

4.3.3 Feedforward Torque Terms. Feedforward torque is the next largest component in the total control torque. It consists of three components based on weighted errors and angular positions. The error and position terms have time delays ranging from 0 to 3 sample periods. All three proportional feedforward torque parameters, F_{0P} , F_{1P} , and F_{2P} , were selected before the integral parameters. Again, the tradeoff in selecting sets of these parameters was the balance between reduced initial errors versus endpoint oscillations. The first two parameters produced improvements in manipulator tracking. The third term, F_{2P} , only acted to increase oscillations. It was set to zero. The λ and β terms were re-adjusted and followed by a re-evaluation of F_{0P} and F_{1P} .

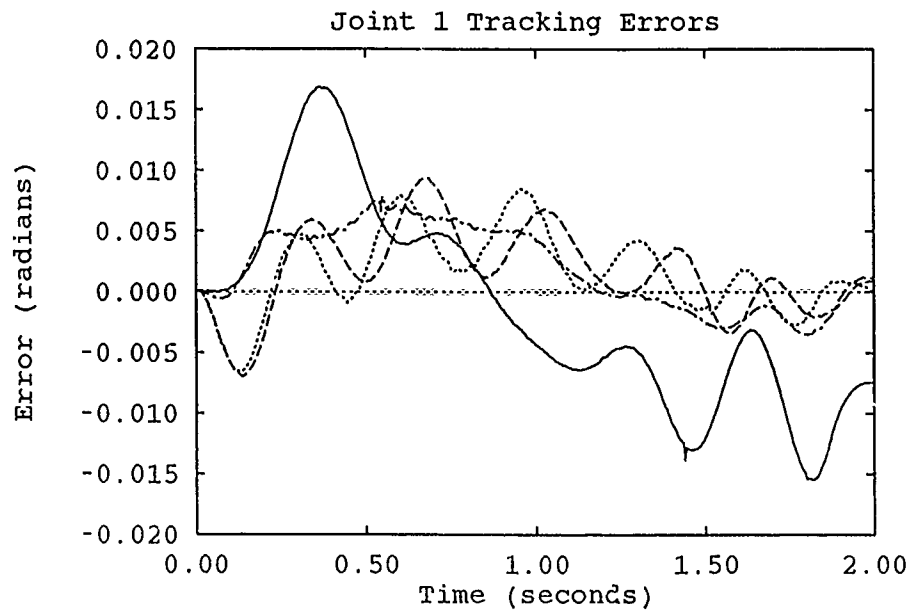


Figure 4.5. Error Plots During Tuning Tarokh's Algorithm

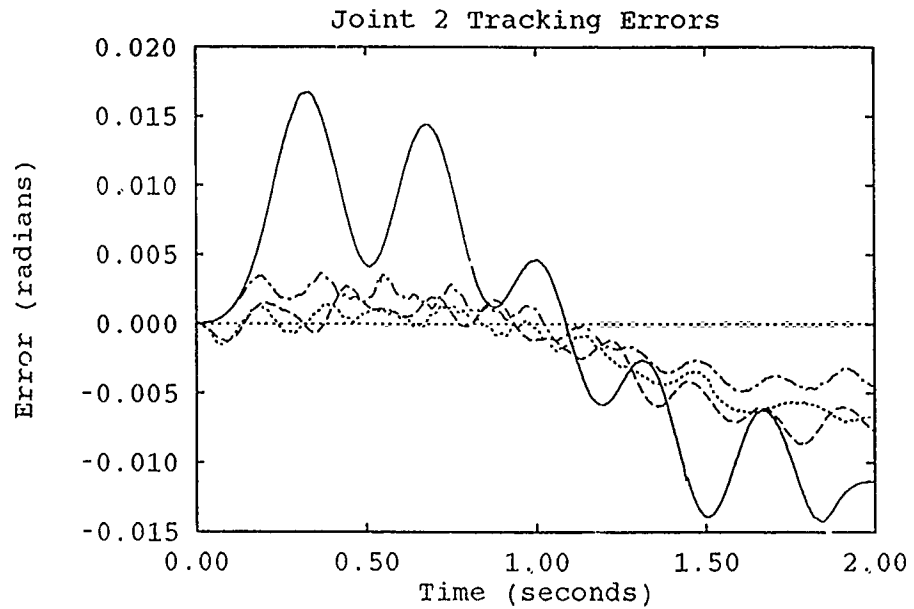


Figure 4.6. Error Plots During Tuning Tarokh's Algorithm

—	After selecting β_P and β_I	After selecting F_{1P}
---	After selecting F_{1I}	- - - - -	After selecting F_{0I} and F_{0P}

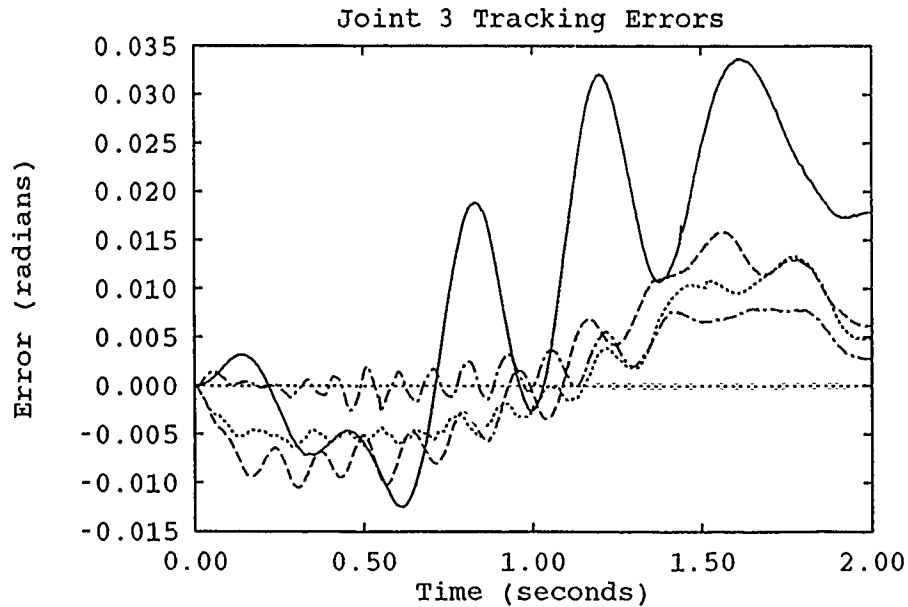


Figure 4.7. Error Plots During Tuning Tarokh's Algorithm

—	After selecting β_P and β_I	After selecting F_{1P}
----	After selecting F_{1I}	- - - -	After selecting F_{0I} and F_{0P}

Next, the integral feedforward parameters, F_{0I} , F_{1I} , and F_{2I} , were addressed. As in the case of the proportional parameters, only the first two terms were effective. The third was set to zero. Initial conditions for the three feedforward components did not improve performance. All previously selected parameters were re-adjusted for optimum response.

4.3.4 Feedback Torque Terms. Although these terms (E_{1P} , E_{1I} , E_{2P} , and E_{2I}) are included in the derivation and demonstration of this algorithm, they did not influence the tracking and endpoint accuracy even when set as high as 10^6 . These terms and the feedback torque initial conditions were set to zero.

4.4 Examination of Seraji's Algorithm

This section is a cursory glance at the Seraji's Lyapunov-based decentralized adaptive control algorithm [20] that was discussed in Chapter 2. Leahy also experimented with this algorithm [10] and expressed this controller as being divided into feedforward (τ_{ff}),

Table 4.2. Design Parameters Selected During Tuning Tarokh's Algorithm

	Parameters Under Consideration:		
	β_P and β_I	F_{0P} and F_{1P}	F_{0I} and F_{1I}
Constants:			
α	diag[1,1,1]	diag[1,1,1]	diag[1,1,1]
Auxiliary Terms:			
β_I	diag[10,10,2]	diag[5,5,10]	diag[5,5,10]
β_P	diag[1000,3000,400]	diag[1600,4500,700]	diag[1600,4500,750]
Eigenvalues:			
λ_1	diag[-7,-.7,-.7]	diag[-.85,-.85,-.85]	diag[-.85,-.85,-.85]
λ_2	diag[-.95,-.95,-.95]	diag[-.9,-.9,-.9]	diag[-.95,-.95,-.95]
Feedforward Terms:			
F_{0F}	diag[0,0,0]	diag[1200,1000,500]	diag[1050,850,525]
F_{1P}	diag[0,0,0]	diag[900,700,450]	diag[1400,1200,900]
F_{0I}	diag[0,0,0]	diag[0,0,0]	diag[11,8,9]
F_{1I}	diag[0,0,0]	diag[0,0,0]	diag[5,5,5]
F_{2P} and F_{2I}	diag[0,0,0]	diag[0,0,0]	diag[0,0,0]
Feedback Terms:			
E_{1P} and E_{2P}	diag[0,0,0]	diag[0,0,0]	diag[0,0,0]
E_{1I} and E_{2I}	diag[0,0,0]	diag[0,0,0]	diag[0,0,0]
Initial Conditions:			
$\eta(0)$	diag[10,30,-7]	diag[10,30,-7]	diag[10,30,-7]

feedback (τ_{fb}), and auxiliary (τ_{ax}) torque components.

$$\tau = \tau_{ax} + \tau_{ff} + \tau_{fb} \quad (4.8)$$

where:

$$\tau_{ff} = C\theta_d + B\dot{\theta}_d + A\ddot{\theta}_d \quad (4.9)$$

$$\tau_{fb} = K_v\dot{e} + K_p e \quad (4.10)$$

θ_d is the desired trajectory and $e = (\theta_d - \theta)$. A , B , C , K_v , and K_p are all $n \times n$ adaptive matrices [10]. These equations are digitally implemented by:

$$\begin{aligned}\tau_{ax}(k) = & \tau_{ax}(k-1) + \alpha(1)\frac{T_s}{2}[\mathbf{R}(k) + \mathbf{R}(k-1)] \\ & + \beta(1)[\mathbf{R}(k) - \mathbf{R}(k-1)]\end{aligned}\quad (4.11)$$

$$\begin{aligned}\mathbf{K}_p(k) = & \mathbf{K}_p(k-1) + \alpha(2)\frac{T_s}{2}[\mathbf{R}(k)\mathbf{e}(k) + \mathbf{R}(k-1)\mathbf{e}(k-1)] \\ & + \beta(2)[\mathbf{R}(k)\mathbf{e}(k) - \mathbf{R}(k-1)\mathbf{e}(k-1)]\end{aligned}\quad (4.12)$$

$$\begin{aligned}\mathbf{K}_v(k) = & \mathbf{K}_v(k-1) + \alpha(3)\frac{T_s}{2}[\mathbf{R}(k)\dot{\mathbf{e}}(k) + \mathbf{R}(k-1)\dot{\mathbf{e}}(k-1)] \\ & + \beta(3)[\mathbf{R}(k)\dot{\mathbf{e}}(k) - \mathbf{R}(k-1)\dot{\mathbf{e}}(k-1)]\end{aligned}\quad (4.13)$$

$$\begin{aligned}\mathbf{A}(k) = & \mathbf{A}(k-1) + \alpha(4)\frac{T_s}{2}[\mathbf{R}(k)\ddot{\theta}_d(k) + \mathbf{R}(k-1)\ddot{\theta}_d(k-1)] \\ & + \beta(4)[\mathbf{R}(k)\ddot{\theta}_d(k) - \mathbf{R}(k-1)\ddot{\theta}_d(k-1)]\end{aligned}\quad (4.14)$$

$$\begin{aligned}\mathbf{B}(k) = & \mathbf{B}(k-1) + \alpha(5)\frac{T_s}{2}[\mathbf{R}(k)\dot{\theta}_d(k) + \mathbf{R}(k-1)\dot{\theta}_d(k-1)] \\ & + \beta(5)[\mathbf{R}(k)\dot{\theta}_d(k) - \mathbf{R}(k-1)\dot{\theta}_d(k-1)]\end{aligned}\quad (4.15)$$

$$\begin{aligned}\mathbf{C}(k) = & \mathbf{C}(k-1) + \alpha(6)\frac{T_s}{2}[\mathbf{R}(k)\theta_d(k) + \mathbf{R}(k-1)\theta_d(k-1)] \\ & + \beta(6)[\mathbf{R}(k)\theta_d(k) - \mathbf{R}(k-1)\theta_d(k-1)]\end{aligned}\quad (4.16)$$

$$\mathbf{R}(k) = \mathbf{W}_p\mathbf{e}(k) + \mathbf{W}_v\dot{\mathbf{e}}(k)\quad (4.17)$$

Since this algorithm is defined as a decentralized controller, all matrices are forced to be diagonal. The diagonal δ , α , and γ matrices used in Equations (2.34-2.36) were implemented as constants and gathered into a new α vector. Similarly, the corresponding \mathbf{p} , β , and λ matrices are now expressed as a new β vector. These α and β vectors and the diagonal matrices \mathbf{W}_p and \mathbf{W}_v are operator-specified constants.

Again, as with Tarokh's algorithm, the torque components were ranked in descending order of magnitude, assuming the presence of small joint angle and velocity errors, to determine which parameters should contribute the most to developing the torques for our particular controller. The order of these components is the same as the corresponding portions of Tarokh's controller:

1. Auxiliary torque - This term is the weighted sum (and difference) of the two most recent weighted errors \mathbf{R} and is only first order as far as the error terms are concerned.

2. Feedforward torques - The three feedforward torque components are products of first order error terms and second order angular position, velocity, and acceleration terms. The position errors are assumed to be small while the position terms can be as large as $\pm\pi$. High instantaneous acceleration and velocity terms must also be regarded with caution. This may be why Seraji chose to omit the feedforward terms from his implementation of the algorithm in [20].
3. Feedback torques - As in Tarokh's algorithm, these torques are third order products of angular property errors. For the desired small errors, the multiplicative products of the parameters associated with these terms may need to be extremely large to produce any noticeable contribution.

4.4.1 Differences in Tarokh's and Seraji's Algorithms. There are two major differences in these two decentralized adaptive controllers other than the stability theorem used. One has to do with the trajectory components used. The other concerns the nature of the adaptive matrices.

Tarokh's controller, Equation (4.1), is formulated on only angular position and position error terms. All of the torque components and adaptive parameters are based on functions of position and position error. Seraji's controller, Equations (4.8-4.10), uses error terms and desired trajectory attributes. The adaptive matrices depend on position and velocity errors contained in the weighting matrix \mathbf{R} . These velocity errors, which are calculated by $\dot{e}(k) = [c(k) - e(k-1)]/T_s$, where $e(k) = \theta_d(k) - \theta(k)$, are essentially a weighted position error term. The feedforward torque components and their respective adaptive matrices rely on desired trajectory components which avoids the use of noisy acceleration measurements.

In Seraji's algorithm, as implemented by both Seraji [20] and Leahy [10], only the weighted error terms allow the matrices to adapt on a joint by joint level. The α and β terms are constants applied to a particular matrix. The adaptive elements of Tarokh's controller are allowed the additional freedom of diagonal matrices \mathbf{E} and \mathbf{F} to aid in the adaptation of the matrices used in the feedforward and feedback torque terms. Both controllers express the auxiliary term as a linear function of weighted error terms.

4.4.2 Experimental Runs Made Using Seraji's Algorithm. The set of tuning parameters used by both Seraji [20] and Leahy [10] is given in Table 4.3. All initial conditions were set to zero. These parameters were used to run all proposed test trajectories listed in Table 3.1. These parameters reduced the controller to using only auxiliary and feedback torques. Setting $\alpha(3) - \alpha(6) = 0$ deleted the feedforward torque terms. Setting the β terms all to zero removed the contribution of the terms associated with the differences of the two most recent weighted error terms. Neither Seraji nor Leahy tuned the controller for optimum response. None of the experimental results are particularly impressive. Nor is it surprising that the plots contained in Appendix Q illustrate that this controller is sensitive to initial positions, the signs of desired trajectory parameters, and payload.

Table 4.3. Seraji's Design Parameters

Auxiliary terms:	$\alpha(1) = 50$	$\beta(1) = 0$
Feedback terms:	$\alpha(2) = 100$	$\beta(2) = 0$
	$\alpha(3) = 800$	$\beta(3) = 0$
Feedforward terms:	$\alpha(4) = 0$	$\beta(4) = 0$
	$\alpha(5) = 0$	$\beta(5) = 0$
	$\alpha(6) = 0$	$\beta(6) = 0$
Weighted Error terms:	$W_p(1) = 30$	$W_v(1) = 20$
	$W_p(2) = 40$	$W_v(2) = 20$
	$W_p(3) = 12$	$W_v(3) = 4$

4.4.3 Evaluation of Tarokh and Seraji Controllers. The decentralized adaptive controllers will be evaluated on their ability to respond to conditions other than their tuning trajectory conditions. Is this controller sensitive to payload, trajectory speed, or different trajectories?

Tarokh's algorithm was tested over the trajectories listed in Table 3.1. The controller was tuned for Trajectory 0. The test trajectories demonstrate the response of the manipulator to scenarios other than the tuning conditions. Figures 4.8-4.10 show how the Tarokh and Seraji decentralized adaptive control schemes match up against each other. Appendix Q contains the set of plots comparing all trajectories. Appendix P contains plots which

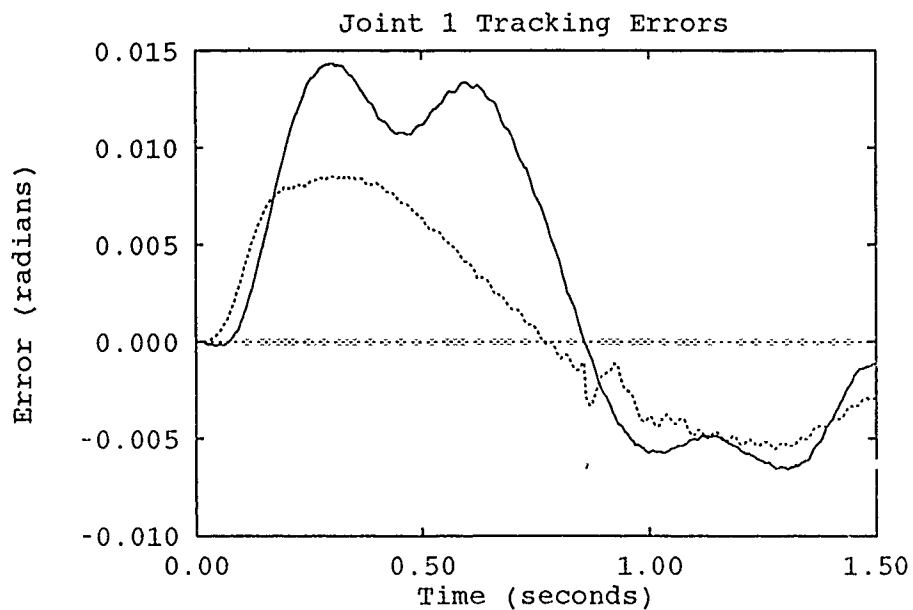


Figure 4.8. Comparison of Tarokh's and Seraji's Control Algorithms - Trajectory 1

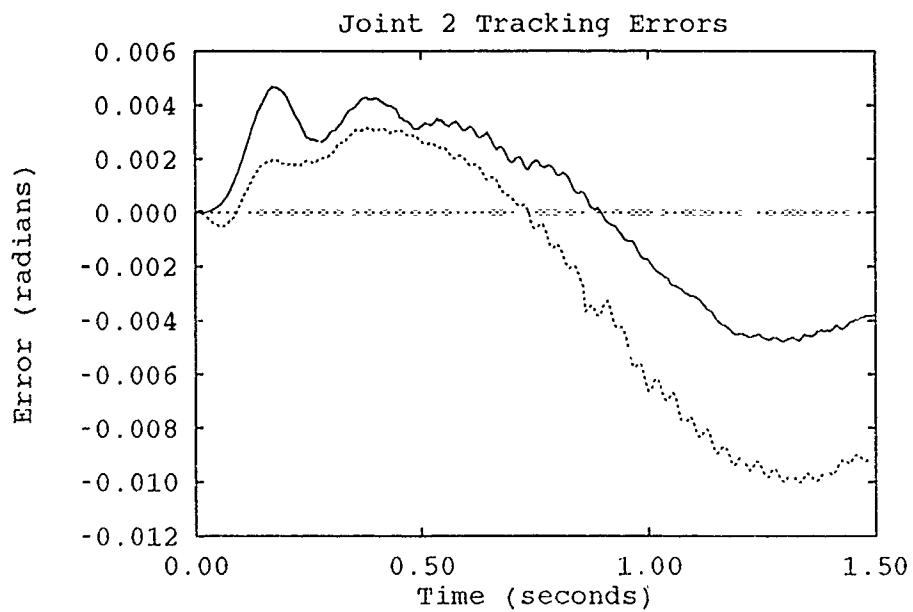
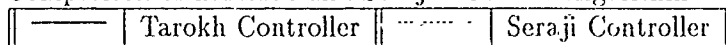


Figure 4.9. Comparison of Tarokh's and Seraji's Control Algorithms - Trajectory 1



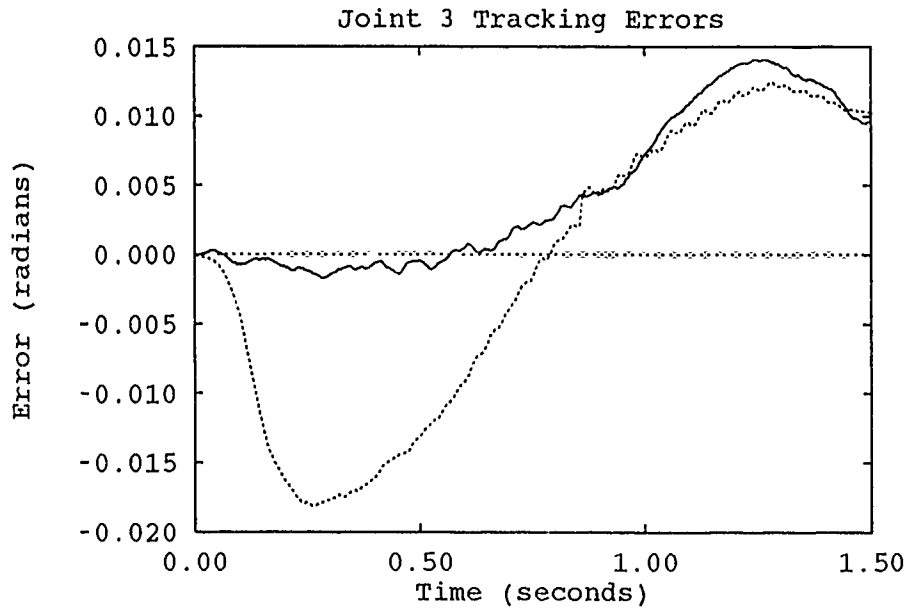
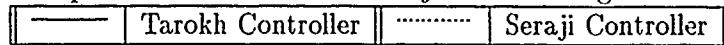


Figure 4.10. Comparison of Tarokh's and Seraji's Control Algorithms - Trajectory 1



compare the error plots generated from the tuning Trajectory 0 to those for the other test trajectories using the same design parameters. Trajectories 0, 2, and 3 all experience movement of $(95^\circ, 45^\circ, -105^\circ)$ in 2.0 seconds and are subject to identical desired velocity and acceleration commands. Tracking errors for Trajectories 2 and 3 are larger than those for Trajectory 0. Joints 2 and 3 (especially 3) go into oscillation during Trajectory 2. Figures 4.11-4.13 illustrate the Tarokh controller's sensitivity to position terms.

Trajectories 4 and 5 apply identically generated trajectory commands to different initial conditions. When compared to Trajectory 0, the initial positions differ and the desired velocity and acceleration terms should be opposite in sign since the movement of Joints 1 and 2 is reversed. Figures 4.14-4.16 show how poorly the Tarokh controller responds to the combination of different initial conditions and reversed trajectories for Joints 1 and 2. Tarokh's controller is very sensitive to these conditions. Peak tracking errors for Joints 1 and 2 are five times larger for Trajectory 4 than for Trajectory 0. Tarokh's algorithm depends on position and position error terms only. Seraji's controller is sensitive to these terms as well as the desired velocity trajectory components.

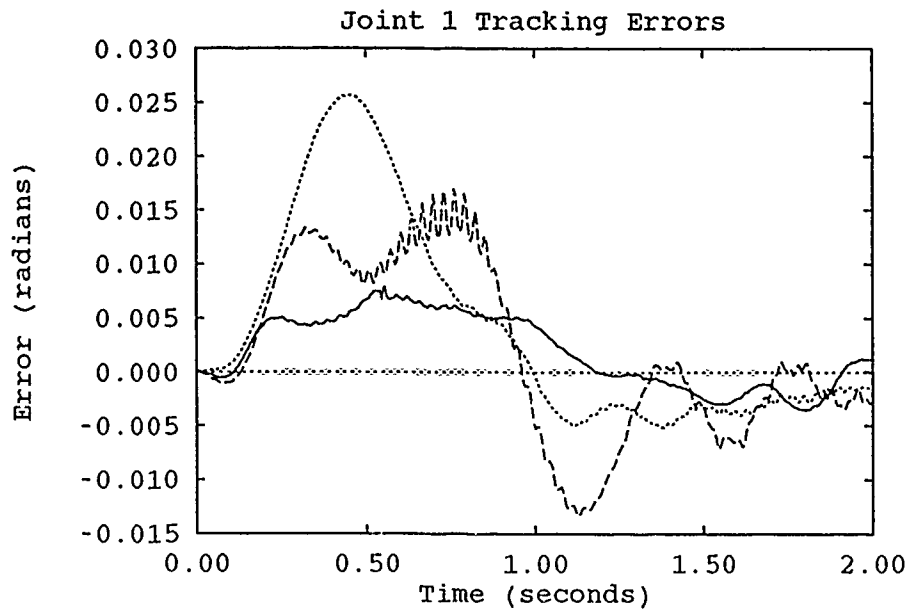


Figure 4.11. Effects of Tarokh Discrete Adaptive Controller's Trajectory 0 Parameters on Trajectories 2 and 3

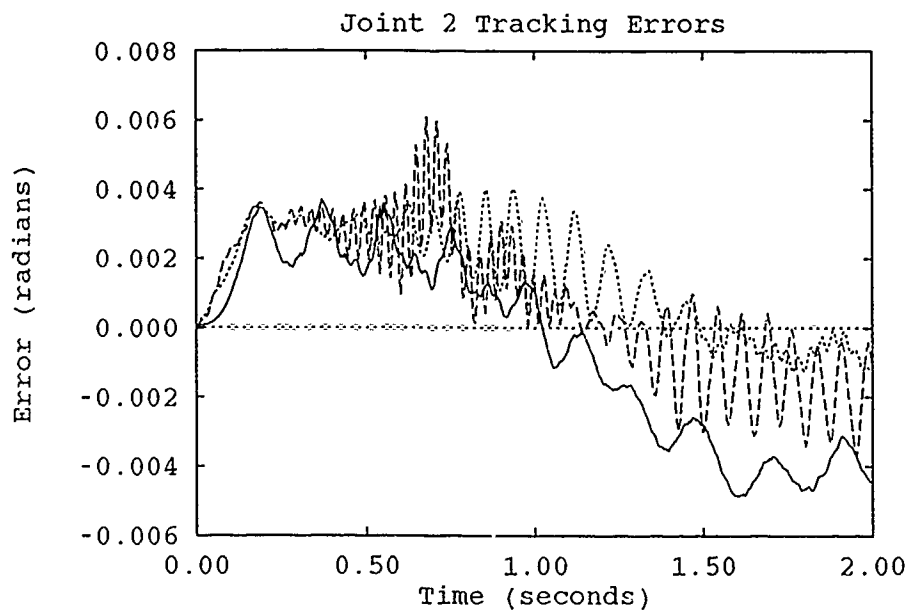


Figure 4.12. Effects of Tarokh Discrete Adaptive Controller's Trajectory 0 Parameters on Trajectories 2 and 3

—	Trajectory 0
- - -	Trajectory 2
...	Trajectory 3

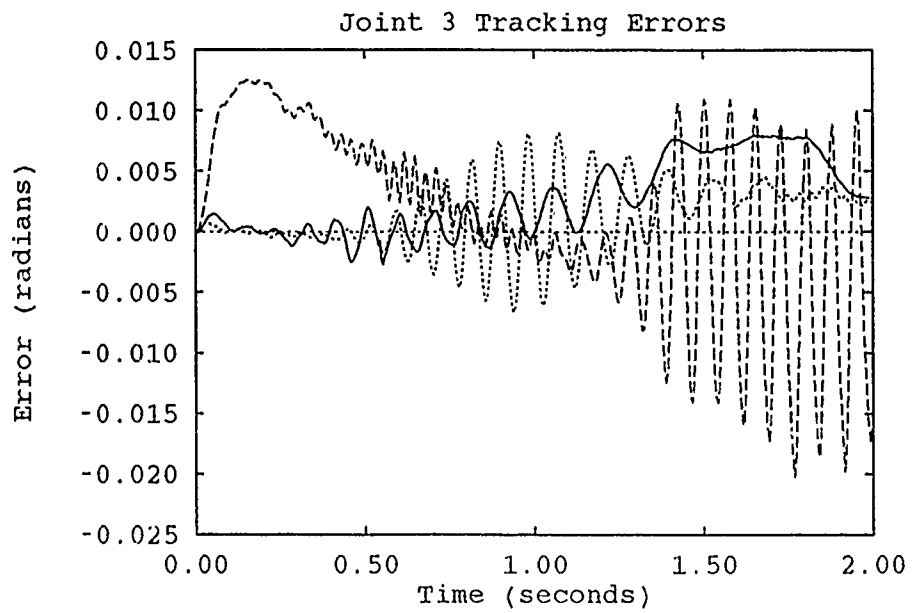


Figure 4.13. The Effects of Trajectory 0 Tuning Parameters on Trajectories 2 and 3

—	Trajectory 0	Trajectory 3
----	Trajectory 2		

The movement of Trajectory 6 is the same as Trajectory 1 except Joint 3 moves in the opposite direction. Desired velocity and acceleration terms for Joint 3 should have opposite signs. Figures 4.17-4.19 illustrate the Tarokh controller's inability to handle changes in speed, coupled with reversed trajectory for Joint 3. Figures 4.20-4.22 demonstrate the effects of trajectory speed and payload on a Tarokh controller. Trajectory 1 causes the same angular position changes as Trajectory 0 but in 1.5 seconds instead of 2.0 seconds. The desired trajectory components will all be changed. Tarokh's algorithm appears to be unable to provide trajectory tracking control for conditions other than the specific tuning trajectory.

4.5 Summary

This chapter examined the decentralized digital adaptive controllers of Tarokh [22] and Seraji [20]. They are proposed as computationally simple, fast controllers. Each controller is mathematically developed as an algorithm that uses its parameters to develop a control torque as the sum of auxiliary, feedforward, and feedback torques. Both algorithms are heavily influenced by auxiliary torque terms. In the tuning process, both controllers tend to minimize the need for one of the other torques. Part of the simplicity of these controllers is their independence from the robot's dynamics models, which makes Tarokh's controller exceedingly hard to tune and trajectory dependent.

These controllers did not perform well enough in this study to be considered as an algorithm suitable for gross motion control but, the inclusion of the above mentioned auxiliary torques into Model Based Adaptive Input Controllers, as discussed in the next chapter, is promising.

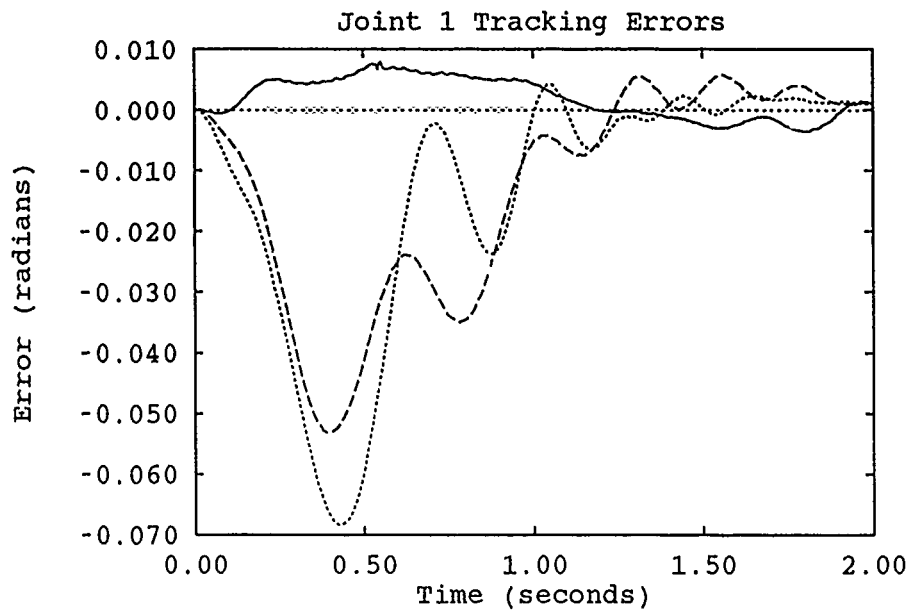


Figure 4.14. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5

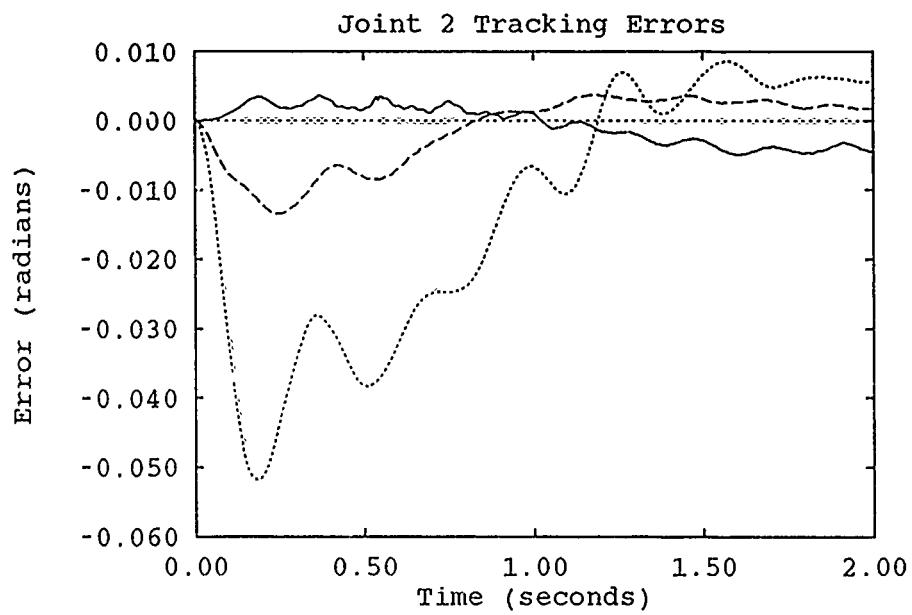


Figure 4.15. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5

—	Trajectory 0	Trajectory 5
- - -	Trajectory 4		

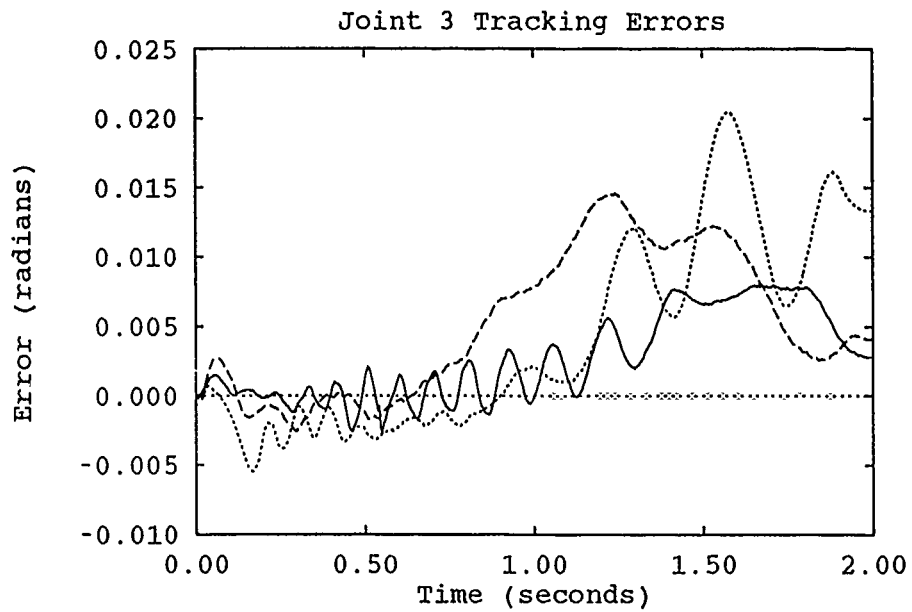


Figure 4.16. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5

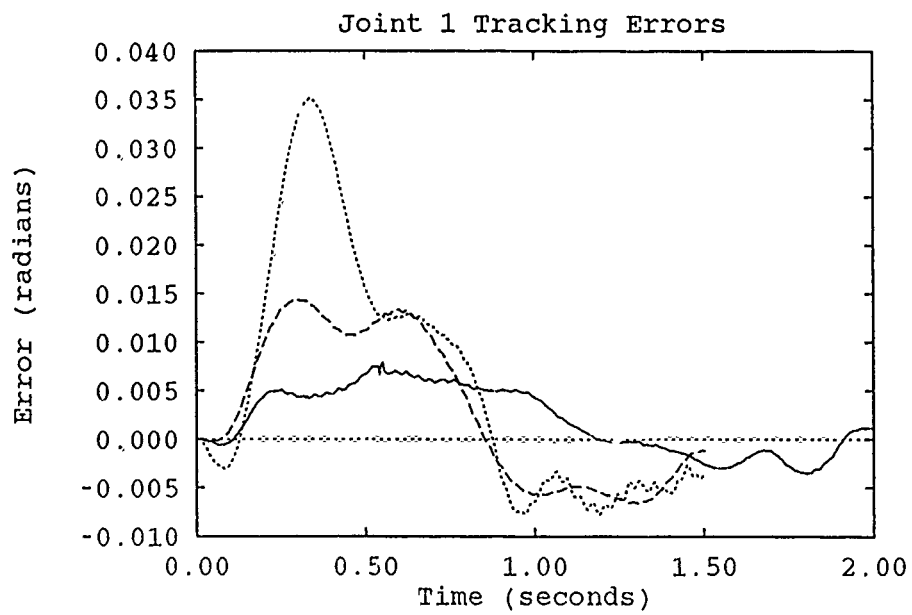


Figure 4.17. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5

—	Trajectory 0	Trajectory 5
---	Trajectory 4		

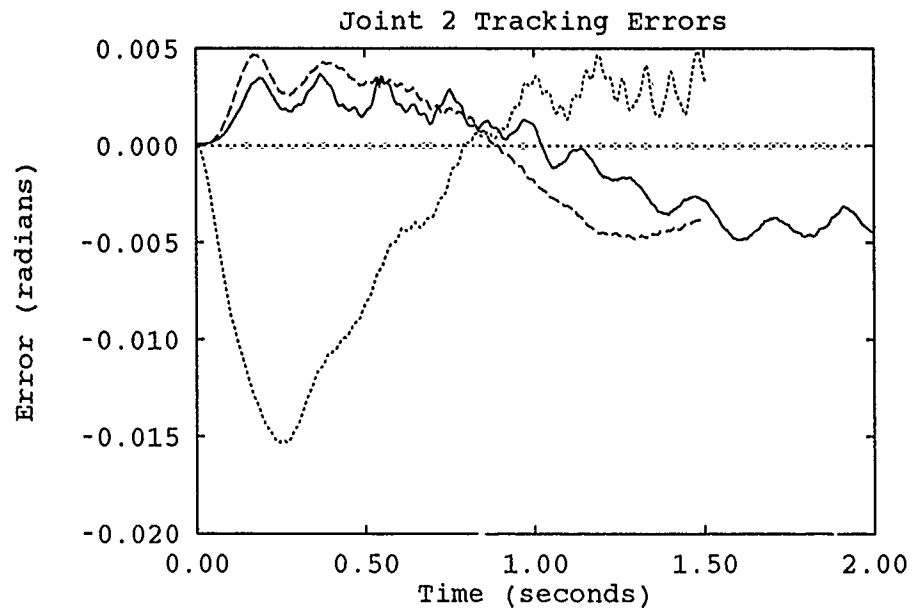


Figure 4.18. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6

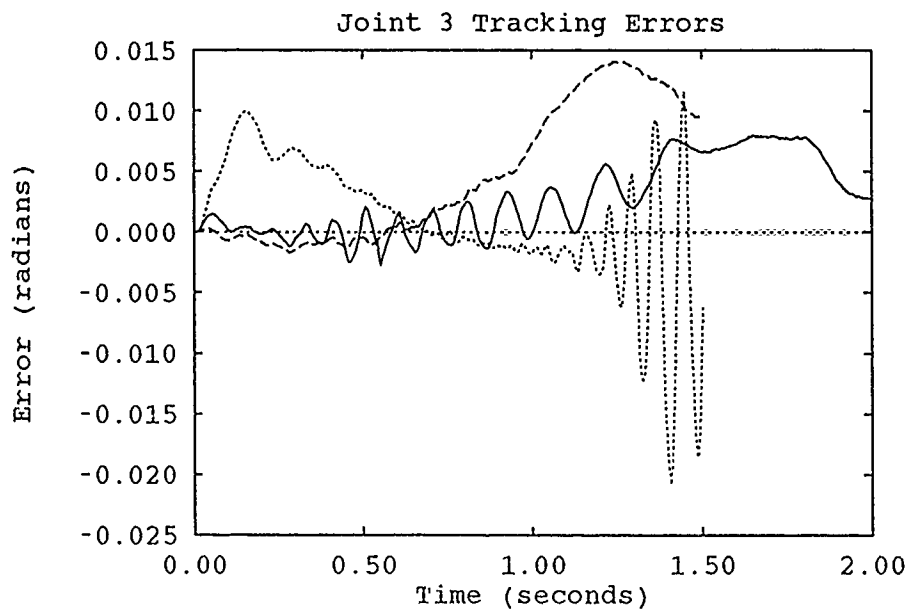


Figure 4.19. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6

—	Trajectory 0	Trajectory 6
---	Trajectory 1		

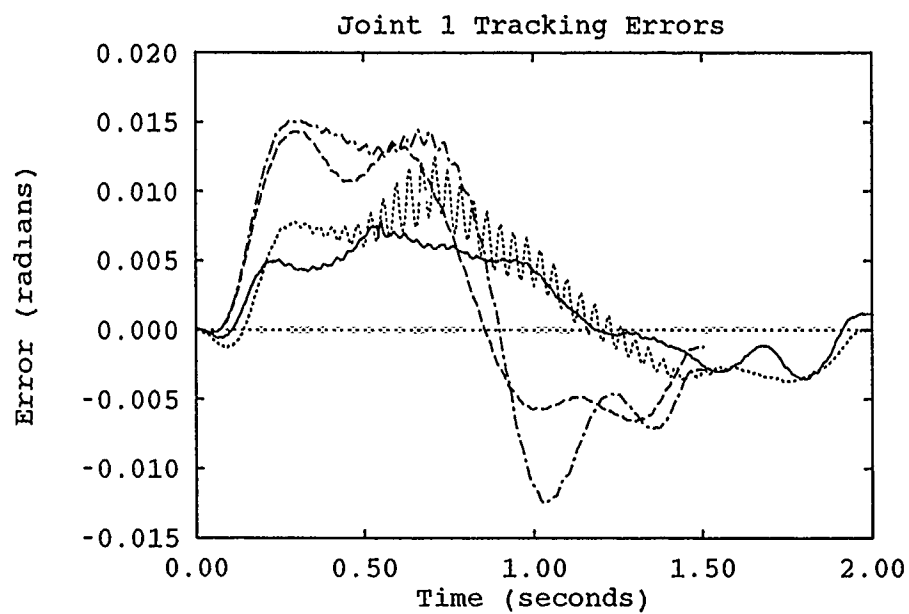


Figure 4.20. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec

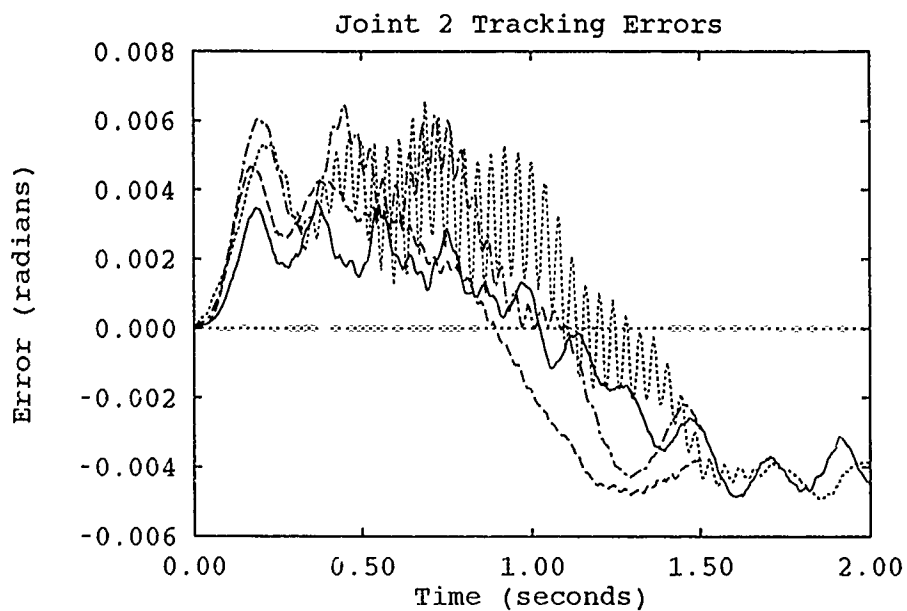


Figure 4.21. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec

—	Trajectory 0	Trajectory 0 with payload
----	Trajectory 1	-.-.-.-	Trajectory 1 with payload

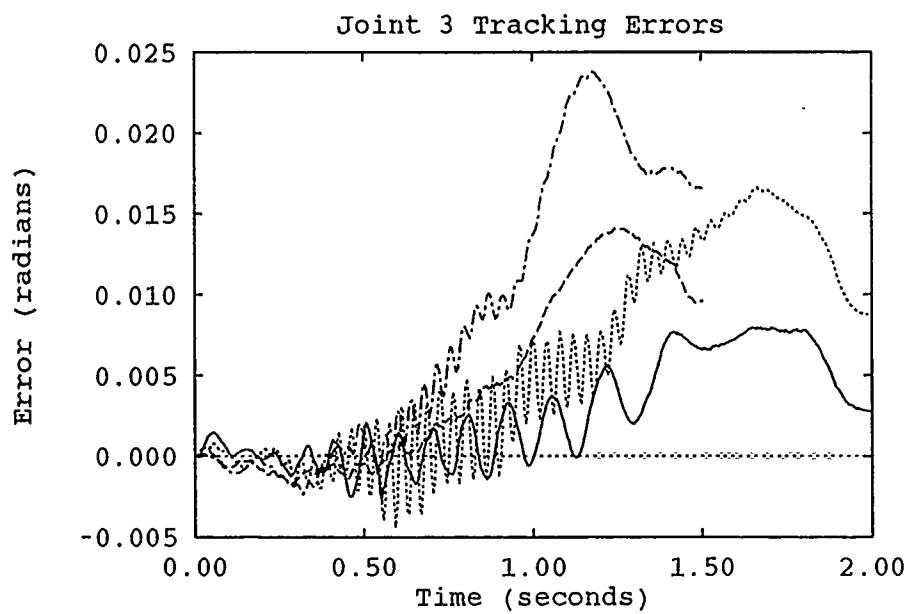


Figure 4.22. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec

—	Trajectory 0	Trajectory 0 with payload
----	Trajectory 1	-.-.-.-	Trajectory 1 with payload

V. Model-Based Auxiliary Input Controllers

5.1 Model-Based Auxiliary Input Controller Implementation

Leahy used the auxiliary torque component of Seraji's controller and formed a Model-Based Auxiliary Input Controller (MBAIC). This controller couples the auxiliary torque with the feedforward and feedback torques developed by an SMBC controller. The MBAIC is expressed by:

$$\tau = \tau_{ax} + \tau_{fb} + \tau_{ff} \quad (5.1)$$

The feedforward torque term is based on compensation of all manipulator system dynamics. The feedback gains are set to the values listed in Chapter 3. The Seraji and Tarokh auxiliary equations implemented are:

$$\tau_{ax} = \mathbf{f}(k) = \mathbf{f}(k-1) + \alpha \frac{T_s}{2} [\mathbf{R}(k) + \mathbf{R}(k-1)] + \beta [\mathbf{R}(k) - \mathbf{R}(k-1)] \quad (5.2)$$

$$\tau_{ax} = \boldsymbol{\eta}(k) = \boldsymbol{\eta}(k-1) + \beta_P \hat{\boldsymbol{\theta}}_e(k) + (\beta_I - \beta_P) \hat{\boldsymbol{\theta}}_e(k-1) \quad (5.3)$$

The role of the auxiliary input has been reduced from eliminating all dynamics disturbances to compensating for variations from modeled dynamics [10]. The Seraji-based MBAIC had the auxiliary torque term, Equation (5.2), rewritten to allow adaptation of the matrices on a joint-by-joint basis. A similar MBAIC was formed using Tarokh's auxiliary torque term, Equation (5.3). Tarokh's auxiliary torque already adapts at the joint level. The Tarokh-based MBAIC was retuned in the same manner as described in Section 4.3.2. The Seraji-based MBAIC was tested using parameters selected by Leahy and Seraji [10, 20]. The auxiliary torques are initialized to zero. The parameters are listed in Table 5.1.

Table 5.1. MBAIC Tuning Parameters [10]

Seraji Based MBAIC	Tarokh Based MBAIC
$\alpha = [50, 100, 100]$	$\alpha = [1, 1, 1]$
$\beta = [0, 0, 0]$	$\beta_P = [2000, 2000, 700]$
$\mathbf{W}_p = [30, 40, 12]$	$\beta_I = [10, 50, 5]$
$\mathbf{W}_v = [20, 20, 4]$	$\lambda_1 = \lambda_2 = [-.3, -.3, -.3]$

5.2 Comparison of the Tarokh and Seraji-Based MBAIC

Is there a difference in the auxiliary torques as they are implemented? The auxiliary torque equations and the weighted errors can be expanded and equivalent terms compared.

$$\begin{aligned}
\mathbf{f}(k) &= \mathbf{f}(k-1) + \alpha \frac{T_s}{2} [\mathbf{R}(k) + \mathbf{R}(k-1)] + \beta [\mathbf{R}(k) - \mathbf{R}(k-1)] \\
&= \mathbf{f}(k-1) + (\alpha \frac{T_s}{2} + \beta) \mathbf{R}(k) + (\alpha \frac{T_s}{2} - \beta) \mathbf{R}(k-1) \\
\mathbf{R}(k) &= \mathbf{W}_p \mathbf{e}(k) + \mathbf{W}_v \dot{\mathbf{e}}(k) \\
&= \mathbf{W}_p \mathbf{e}(k) + \mathbf{W}_v \frac{\mathbf{e}(k) - \mathbf{e}(k-1)}{T_s} \\
&= (\mathbf{W}_p + \frac{\mathbf{W}_v}{T_s}) \mathbf{e}(k) - \frac{\mathbf{W}_v}{T_s} \mathbf{e}(k-1) \\
\eta(k) &= \eta(k-1) + \beta_P \hat{\theta}_e(k) + (\beta_I - \beta_P) \hat{\theta}_e(k-1) \\
\hat{\theta}_e(k) &= \mathbf{R}_2 \theta_e(k-1) + \mathbf{R}_3 \theta_e(k) \\
&= \mathbf{R}_2 \mathbf{e}(k-1) + \mathbf{R}_3 \mathbf{e}(k)
\end{aligned}$$

Inserting the parameters in Table 5.1, the auxiliary equations become:

$$\begin{aligned}
\mathbf{f}(k) &= \mathbf{f}(k-1) + \text{diag}[1003.5, 2010, 4003] \mathbf{e}(k) + \text{diag}[3.75, 10, 3] \mathbf{e}(k-1) \\
&\quad - \text{diag}[1000, 2000, 4000] \mathbf{e}(k-2) \\
\eta(k) &= \eta(k-1) + \text{diag}[2180, 2180, 763] \mathbf{e}(k) - \text{diag}[2277.1, 2233.5, 795.35] \mathbf{e}(k-1) \\
&\quad + \text{diag}[107.3, 105.3, 37.53] \mathbf{e}(k-2)
\end{aligned}$$

Some of these terms are harder than others to select, i.e., \mathbf{R}_3 is a function of Tarokh's λ and α parameters, and T_s is not as easy to manipulate as the other variables. Keeping these two limitations in mind, the respective terms of the weighting matrices and the auxiliary torque equations can be equated.

5.3 Model-Based Auxiliary Input Controller Test Results

Both MBAIC controllers were evaluated over all of the test trajectories. The MBAIC algorithms have a clear tracking advantage over their respective decentralized adaptive

controllers and the SMBC controller. Figures 5.1-5.3 are representative of the improvements gained by using the MBAIC algorithms in place of an SMBC controller. The plots in Appendix R contain the tracking errors produced by the MBAICs and compare the MBAIC to the SMBC used to generate the feedback and feedforward torques for the MBAICs. These plots show that the Tarokh-based MBAIC controller is better than or comparable to the Seraji-based model on all test trajectories. However, as detailed in the preceding section, equivalence of these two controllers is not easy to achieve in view of the origins of some of the multipliers used in the adaptation of the auxiliary terms. The Seraji-based MBAIC may be degraded by changes in the PUMA. The Seraji-based MBAIC was tuned only during Leahy's work [10]. The no-load tracking errors are all less than 3 milliradians. Trajectory 3 is the only loaded test condition where errors exceeded 3 milliradians.

5.4 Comparison of MBAIC and AMBC Controllers

Appendix S contains comparisons of the tracking errors produced by the MBAIC algorithms and the 19-parameter AMBC controller. The AMBC controller works better for all conditions except Trajectory 6, but in several instances the peak errors developed by either trajectory are less than three milliradians. Both control methods provide a high degree of gross manipulator control.

5.5 Summary

This chapter discussed two implementations of an MBAIC controller that used different auxiliary torque equations. The similarity of the auxiliary torque equations was discussed. The tracking performance of these controllers was compared and then compared to SMBC and AMBC algorithms.

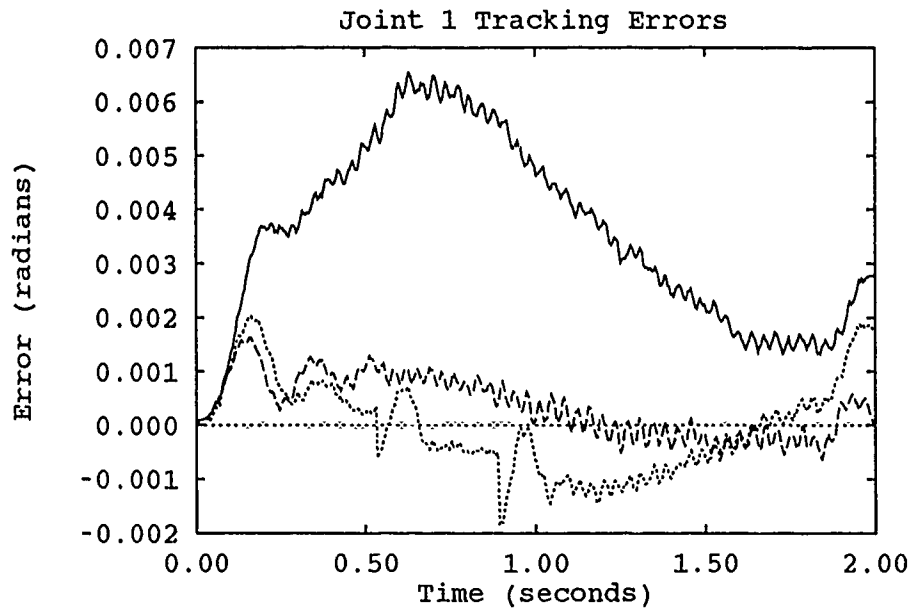


Figure 5.1. Comparison of MBAIC and SMBC Controllers - Trajectory 0

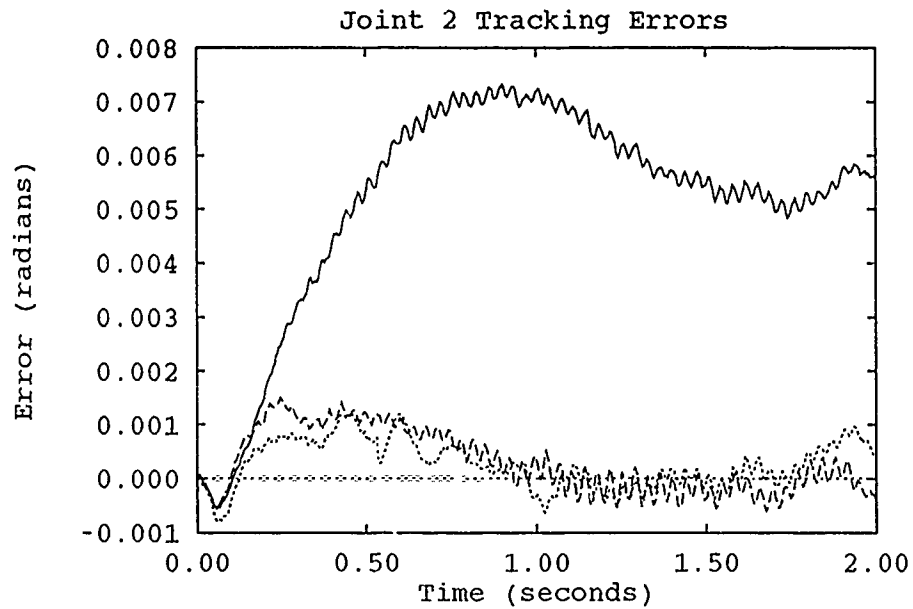


Figure 5.2. Comparison of MBAIC and SMBC Controllers - Trajectory 0

-----	Tarokh-Based MBAIC	————	SMBC Controller
.....	Seraji-Based MBAIC		

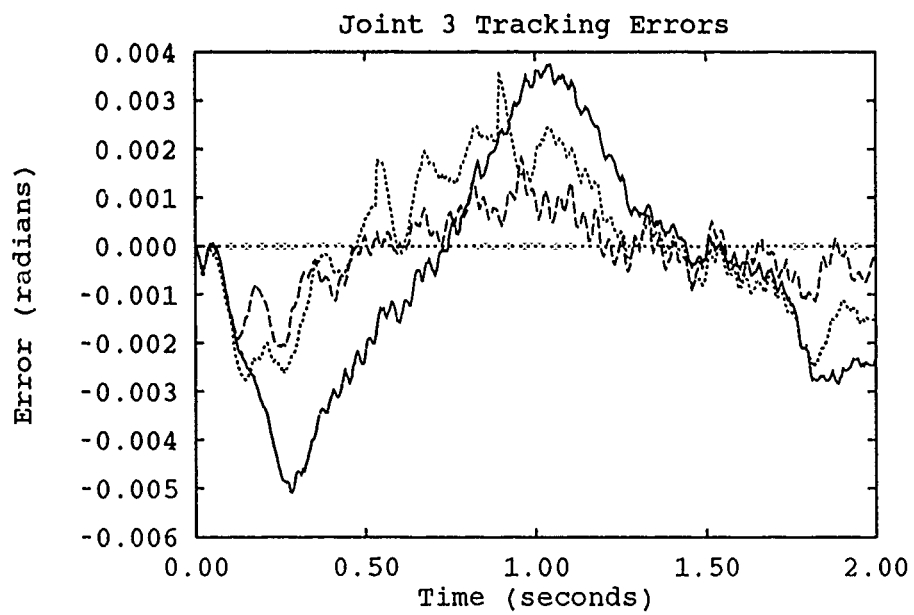


Figure 5.3. Comparison of MBAIC and SMBC Controllers - Trajectory 0

-----	Tarokh-Based MBAIC	———	SMBC Controller
.....	Seraji-Based MBAIC		

VI. Conclusions and Recommendations

6.1 Conclusions

Adaptive parameter techniques have been successfully applied to robotic manipulator control. Adaptive Model-Based Control (AMBC) methods proved viable, as did Model-Based Auxiliary Input Control (MBAIC) techniques. The decentralized adaptive controllers, of which the auxiliary inputs were a component, were not able to approximate the degree of control provided by AMBC or MBAIC procedures.

All three types of controllers were experimentally evaluated for the first three links of the PUMA-560 manipulator. The AMBC method sought to improve control by adapting or calculating a set of manipulator parameters. Our 19-parameter AMBC computes 19 manipulator parameters and leaves 12 at preselected values. When uninitialized tests were conducted, all 31 parameters were set to zero and only the 19 adapting parameters ever contributed to developing the model-based feedforward torques. Initialized tests set all 31 parameters to predetermined values, permit 19 parameters to change, and use all 31 parameters to calculate the feedforward torque. Testing showed that adaptation of uninitialized parameters could not compare to the results attained by using properly initialized parameters. 13- and 16-parameter models were compared to the 19-parameter AMBC. The 13-parameter model was heavily dependent on friction terms and did not operate as well as the other options. The 16-parameter AMBC was already established as an excellent algorithm for gross motion control. The addition of three more friction parameters to form the 19-parameter AMBC did not improve upon the 16-parameter AMBC enough to justify implementation. The AMBC algorithm is more complex and computationally demanding than the other two types of controllers examined.

Both decentralized adaptive controllers adapted six diagonal matrices for use in calculating control torques. However, in practice, three of these matrices were disabled causing Seraji's controller [20] to omit feedforward torques and the Tarokh controller [22, 23] to ignore part of the feedforward and all of the feedback torques. Tarokh's decentralized adaptive controller scaled each joint's adaptation process using diagonal matrices of proportional and integral constants, a trajectory parameter, and a diagonal weighted error

matrix. Seraji's controller uses a single diagonal matrix, the trajectory quantity used in developing the adaptive matrix's torque component (i.e., $C\theta_d$ is a feedforward torque component and θ_d is used in adapting C as in Equation (4.9)), and diagonal weighted error and position matrix. The diagonal matrix is implemented as a scalar. These controllers have fewer characteristics adapting on a joint level than the AMBC controller and do not perform as well as the AMBC algorithm.

The MBAIC controllers use the auxiliary torques produced by the decentralized controllers to supplement a Single Model-Based Controller (SMBC). This restores the overall control torque to its full complement of feedforward, feedback, and auxiliary torque components. These controllers easily outperform the decentralized adaptive controllers and SMBCs from which they are derived. Our MBAIC that was based on Tarokh's auxiliary torque computation provided better tracking than the Seraji version.

The AMBC controller worked better than the MBAIC algorithms for all test conditions except Trajectory 6. In several instances the peak errors developed by either trajectory are less than three milliradians. Both controllers provide a high degree of gross manipulator control. The MBAIC controllers are much simpler to implement and computationally much simpler than the AMBC. If computational loading or sample frequencies are a major concern, MBAIC is the method of choice. For most applications, the AMBC controller should be chosen.

The aim of this research was to evaluate control different methods for implementation as gross robotic motion controllers to meet future Air Force applications. The 16-parameter AMBC was verified as a reasonable approach. Two decentralized adaptive algorithms have been rejected and the potential of Model-Based Auxiliary Input Control has been identified.

6.2 Recommendations

The decentralized adaptive controllers contain unnecessary calculations in view of the number of the mathematically derived components that were not used in either implementation. A sample period of 4.5 milliseconds was used for all of the tests performed in this research effort. This software needs to be rewritten to eliminate avoidable additions and

multiplications of terms that will always be set to zero. These controllers may perform admirably with an increased sampling rate.

Additionally, the AMBC and MBAIC techniques used for adaptive robot control should be compared to other techniques proposed in control literature. This can be accomplished in the experimental environment available in the Air Force Institute of Technology Robotic Systems Laboratory.

Bibliography

1. C. H. An, C. G. Atkeson, J.D. Griffiths, and J. M. Hollerbach. Experimental Evaluation of Feedforward and Computed-Torque Control. *IEEE Trans. on Robotics and Automation*, 5(3):368-372, June 1989.
2. B. Armstrong. Friction: Experimental Determination, Modeling, and Compensation. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, volume 3, pages 1422-1427, 1988.
3. B. Armstrong-Helvoury. Stick-Slip Arising From Stribeck Friction. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, volume 2, pages 1377-1382, 1990.
4. D. Bossert. Design of Pseudo-Continuous Time Quantitative Feedback Theory Robot Controllers. Master's thesis, Air Force Institute of Technology, Air University, December 1989.
5. C. Canudas de Wit and V. Seront. Robust Adaptive Friction Compensation. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, volume 2, pages 1383-1388, 1990.
6. K.S. Fu, R.C. Gonzalez, and C.S.G. Lee. *Robotics: Control, Sensing, Vision and Intelligence*. McGraw-Hill Book Company, New York, 1987.
7. P. K. Khosla and T. Kanade. Experimental Evaluation of Nonlinear Feedback and Feedforward Control Schemes for Manipulators. *Int. J. of Robotics Research*, 7(1):18-28, February 1988.
8. R. Klafter, T. Chmielewski, and M.Negin. *Robot Engineering: An Integrated Approach*. Prentice-Hall Inc, 1st edition, 1986.
9. M. B. Leahy, Jr. Experimental Analysis of Model-Based Puma Robot Control. Technical Report ARSL-89-3, Air Force Inst. of Tech., July 1989. Dept. of Elect. and Comp. Eng.
10. M. B. Leahy, Jr. Model-Based Auxiliary Input Control: Development and Experimental Analysis. In *Proceedings of IEEE Conference on Decision and Control*, pages 3340-46, December 1990.
11. M. B. Leahy Jr., D. E. Bossert, and P. V. Whalen. Robust Model-Based Control: An Experimental Case Study. In *Proceedings of IEEE Conference on Robotics and Automation*, pages 1982-1987, 1990.
12. M. B. Leahy, Jr. and P. V. Whalen. Enhancements to Robotic Manipulator Trajectory Tracking. Technical Report ARSL-90-6, Submitted to the Journal of Robotic Systems, Air Force Institute of Technology, June 1990. Department of Electrical and Computer Engineering.
13. M. B. Leahy, Jr. and P. V. Whalen. Direct Adaptive Control of Industrial Manipulators? In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 1991.

14. M.B. Leahy Jr. and G.N. Saridis. Compensation of Industrial Manipulator Dynamics. *Int. J. of Robot Res.*, 8(4):73-84, August 1989.
15. Professor M. B. Leahy, Jr. Personal interviews. AFIT, Wright-Patterson AFB, OH, January and April 1991.
16. A. Morando, R. Horowitz, and N. Sadegh. Digital Implementation of Adaptive Control Algorithms for Robot Manipulators. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 1656-1662, May 1989.
17. R. Ortega and M. W. Spong. Adaptive Motion Control of Rigid Robots: A Tutorial. In *Proceedings of 27th IEEE Conference on Decision and Control*, pages 1575-1584, 1988.
18. S. Sablan. Multiple Model Adaptive Estimation Techniques for Adaptive Model-Based Robot Control. Master's thesis, Air Force Institute of Technology, Air University, December 1989.
19. N. Sadegh and R. Horowitz. Stability and Robustness Analysis of a Class of Adaptive Controllers for Robotic Manipulators. *International Journal of Robotics Research*, 9(3):74-92, June 1990.
20. H. Seraji. Decentralized Adaptive Control of Manipulators: Theory, Simulation, and Experimentation. *IEEE Transactions on Robotics and Automation*, 5(2):183-201, April 1989.
21. J-J. E. Slotine and W. Li. On the Adaptive Control of Robot Manipulators. *International Journal of Robotics Research*, 6(3):49-59, Fall 1987.
22. M. Tarokh. A Discrete-Time Adaptive Control Scheme for Robot Manipulators. *Journal of Robotic Systems*, 7(2):145-166, April 1990.
23. M. Tarokh. Decentralized Digital Adaptive Control of Robot Motion. In *Proceedings of IEEE Conference on Robotics and Automation*, pages 1410-1415, 1990.
24. S. Tosunoglu and D. Tesar. State of the Art in Adaptive Control of Robotic Systems. *IEEE Transactions on Aerospace and Electronic Systems*, 24(5), September 1988.

Appendix A. *13-Parameter Uninitialized Learning Runs*

This section contain the results of 13-parameter adaptation runs for each test trajectory using uninitialized parameters. Each trajectory was run nine times to allow the controller to adapt.

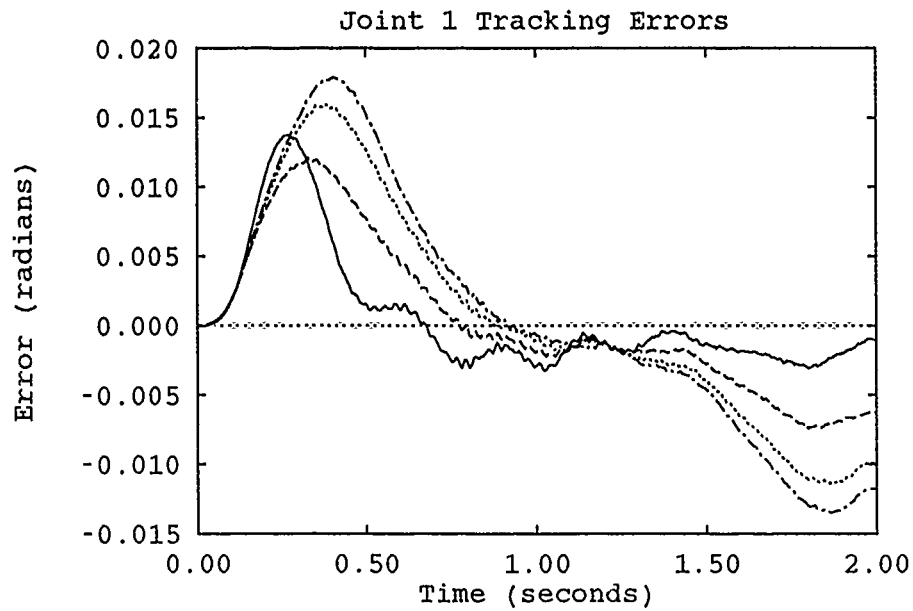


Figure A.1. 13-Parameter Uninitialized Adaptation Runs - Trajectory 2

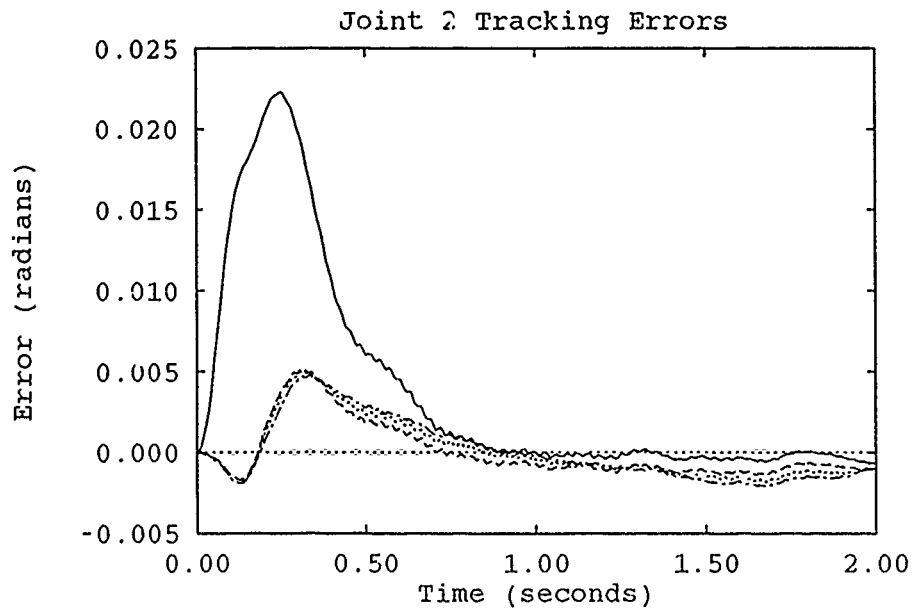


Figure A.2. 13-Parameter Uninitialized Adaptation Runs - Trajectory 2

—	First Run	·····	Sixth Run
---	Third Run	- - - -	Ninth Run

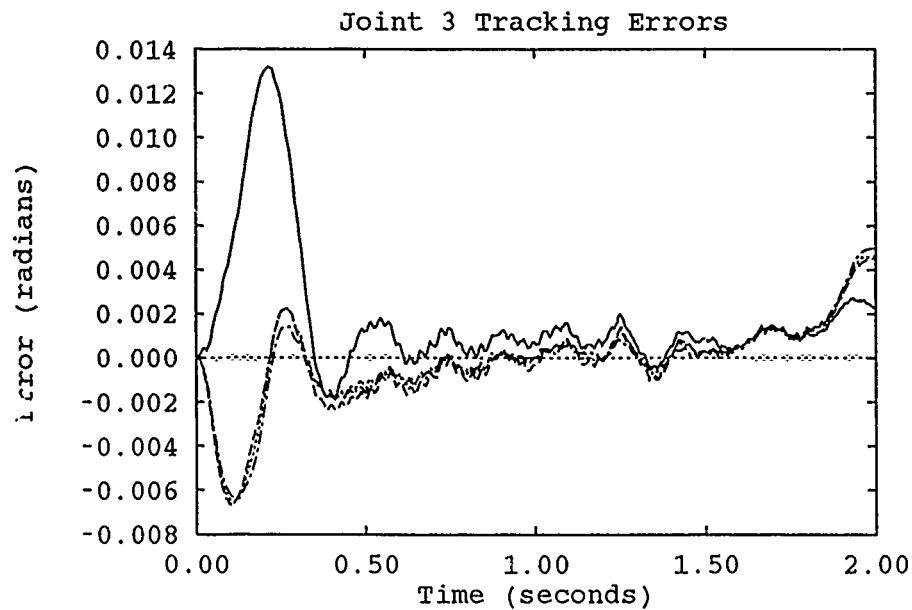


Figure A.3. 13-Parameter Uninitialized Adaptation Runs - Trajectory 2

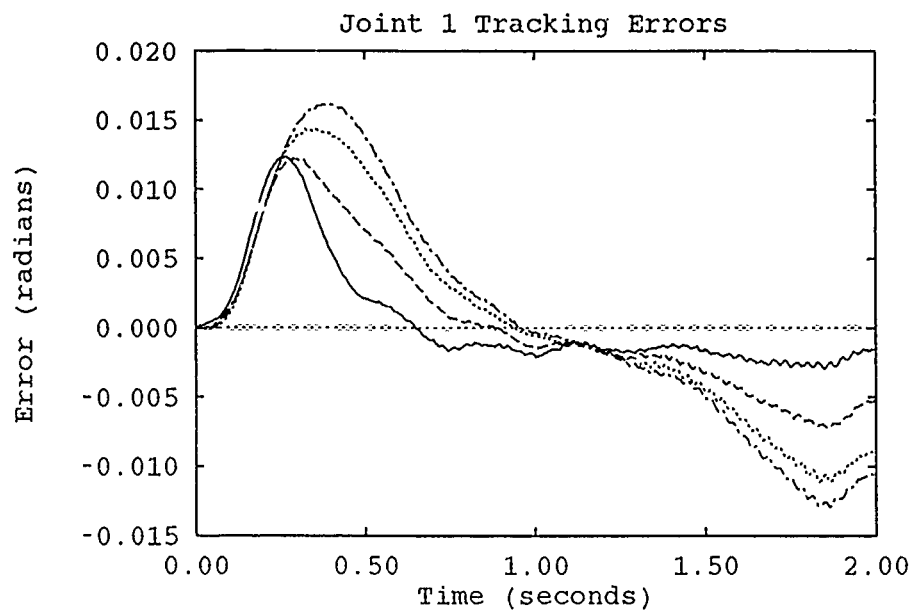


Figure A.4. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3

—	First Run	Sixth Run
---	Third Run	- - - -	Ninth Run

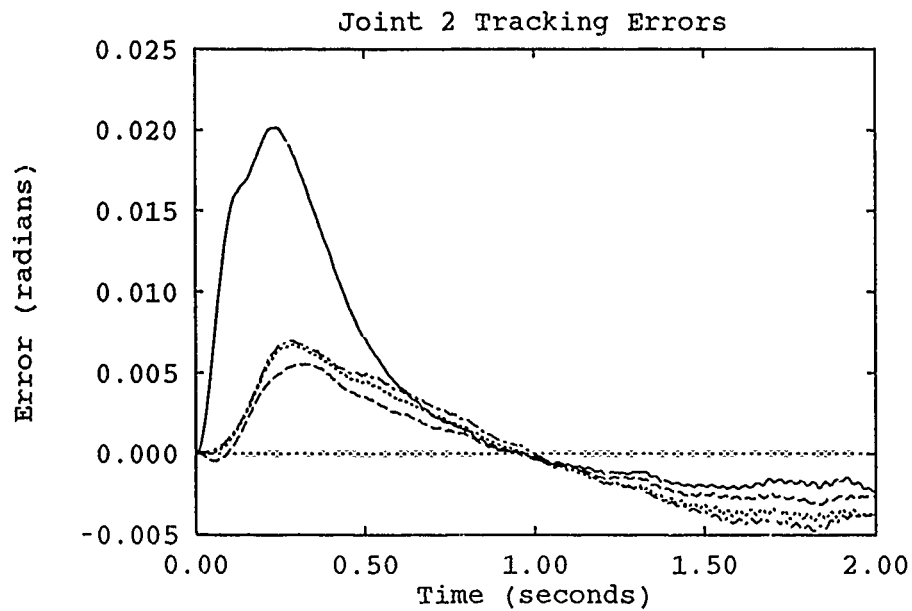


Figure A.5. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3

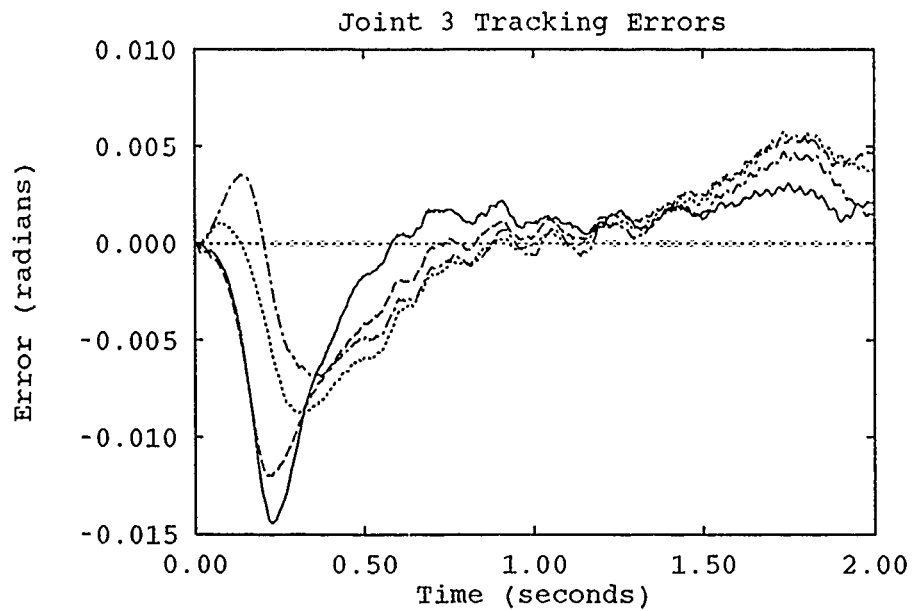


Figure A.6. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3

—	First Run	Sixth Run
- - -	Third Run	- . - .	Ninth Run

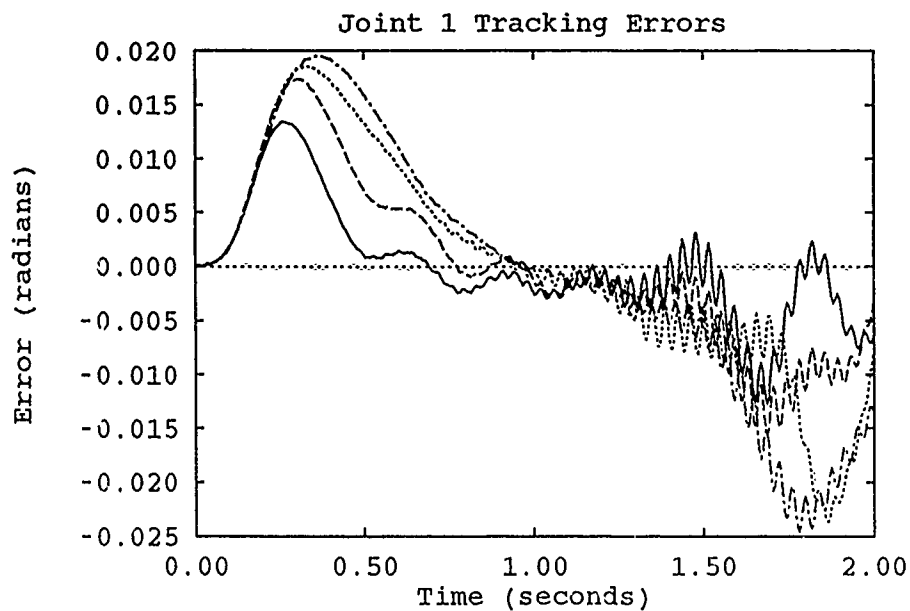


Figure A.7. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

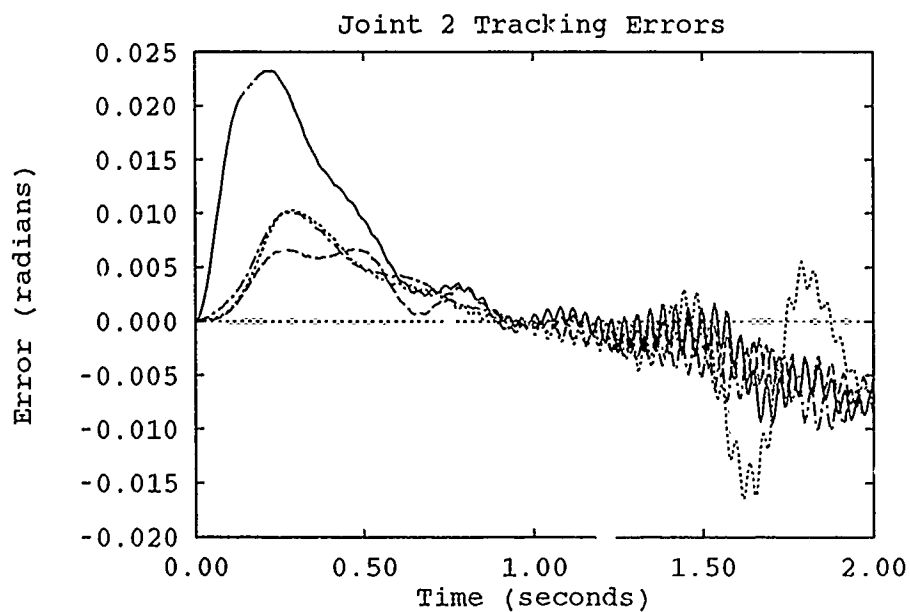


Figure A.8. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

—	First Run	Sixth Run
---	Third Run	- - - -	Ninth Run

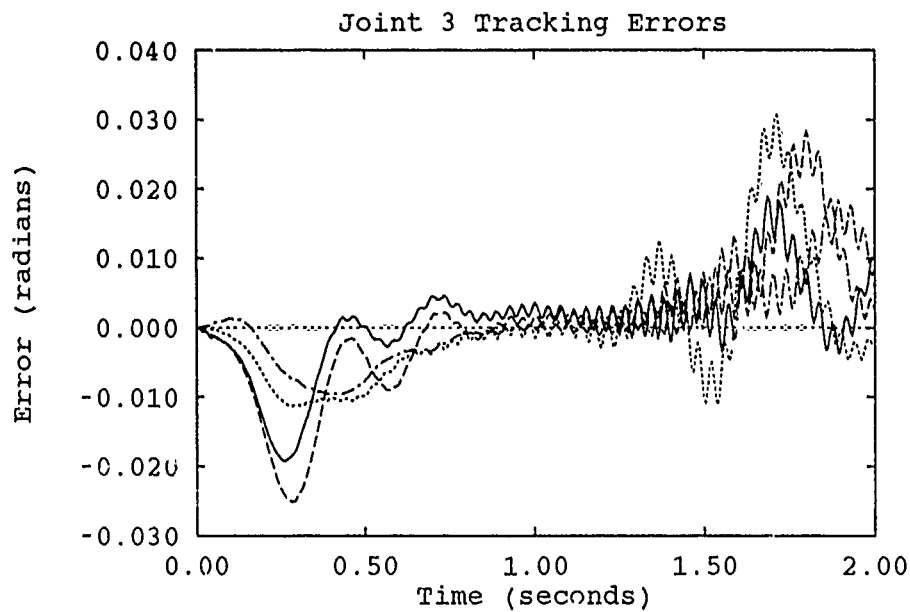


Figure A.9. 13-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

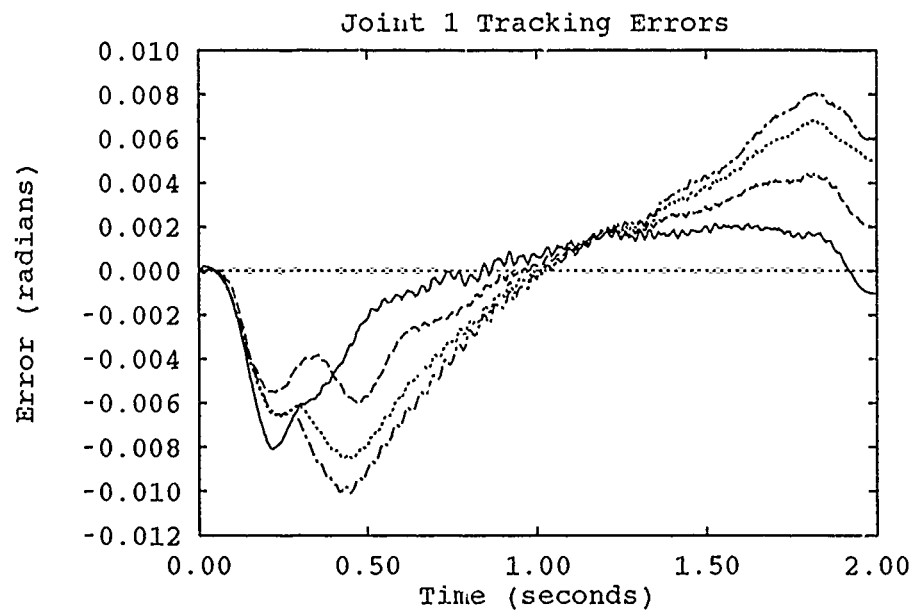


Figure A.10. 13-Parameter Uninitialized Adaptation Runs - Trajectory 4

—	First Run	Sixth Run
---	Third Run	- - - - -	Ninth Run

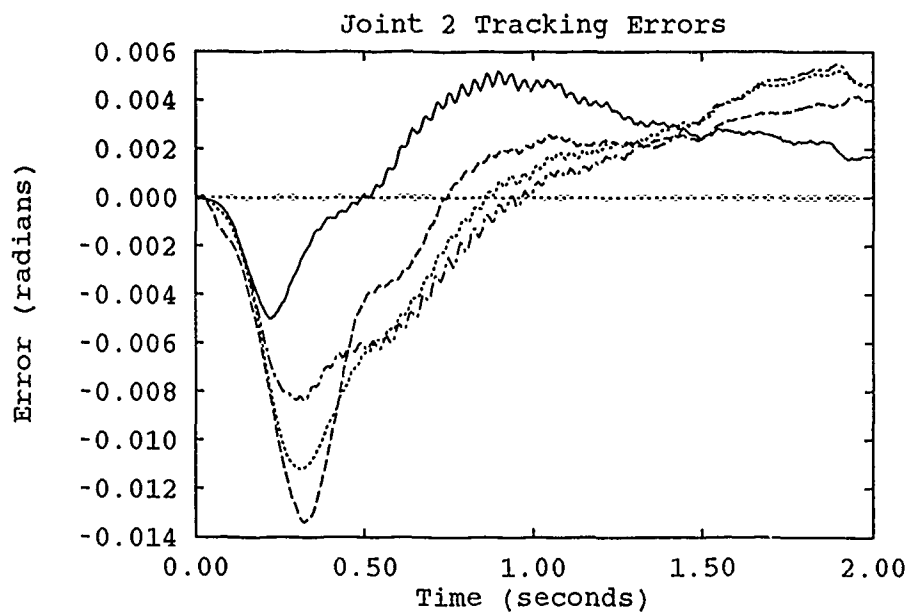


Figure A.11. 13-Parameter Uninitialized Adaptation Runs - Trajectory 4

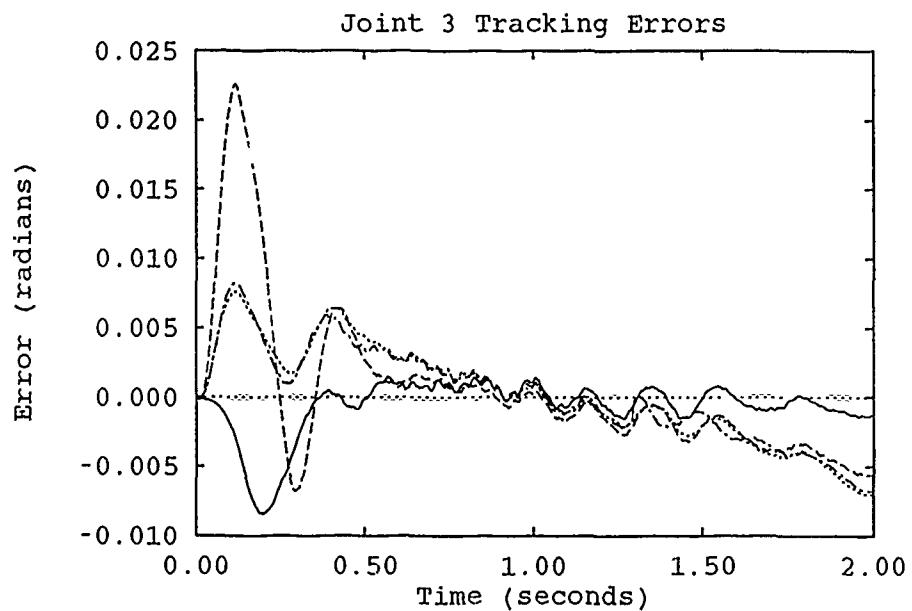


Figure A.12. 13-Parameter Uninitialized Adaptation Runs - Trajectory 4

—	First Run	Sixth Run
- - -	Third Run	- . - .	Ninth Run

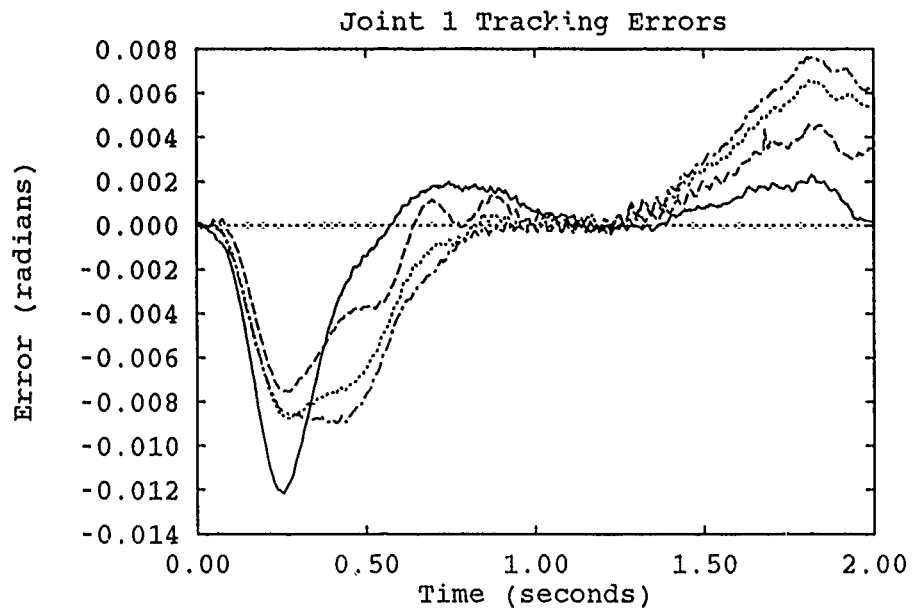


Figure A.13. 13-Parameter Uninitialized Adaptation Runs - Trajectory 5

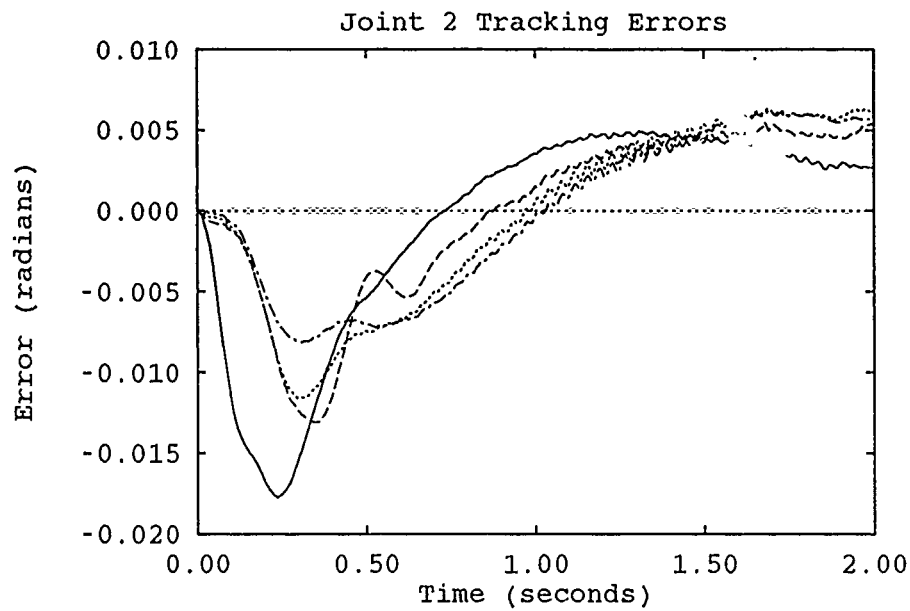


Figure A.14. 12-Parameter Uninitialized Adaptation Runs - Trajectory 5

—	First Run	Sixth Run
----	Third Run	-.-.-.-	Ninth Run

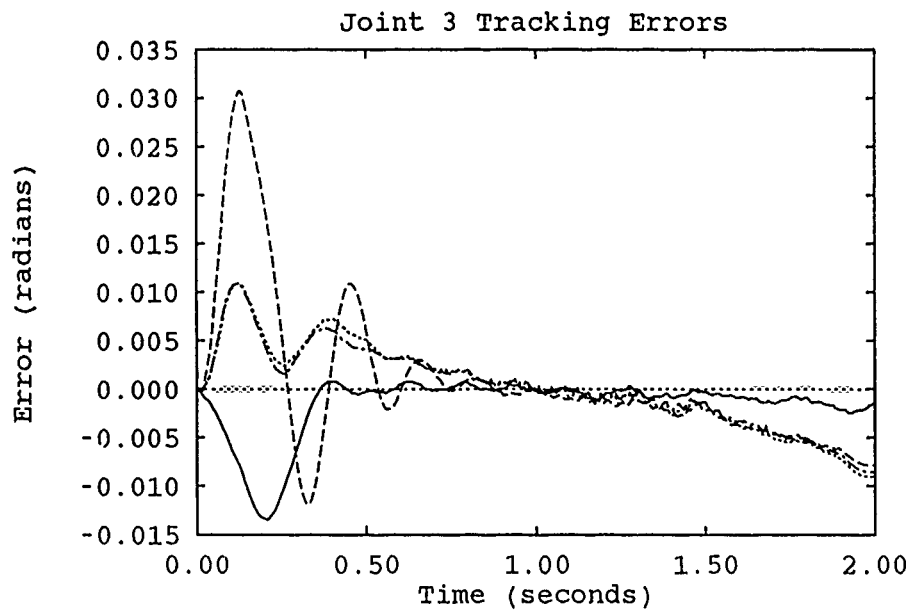


Figure A.15. 13-Parameter Uninitialized Adaptation Runs - Trajectory 5

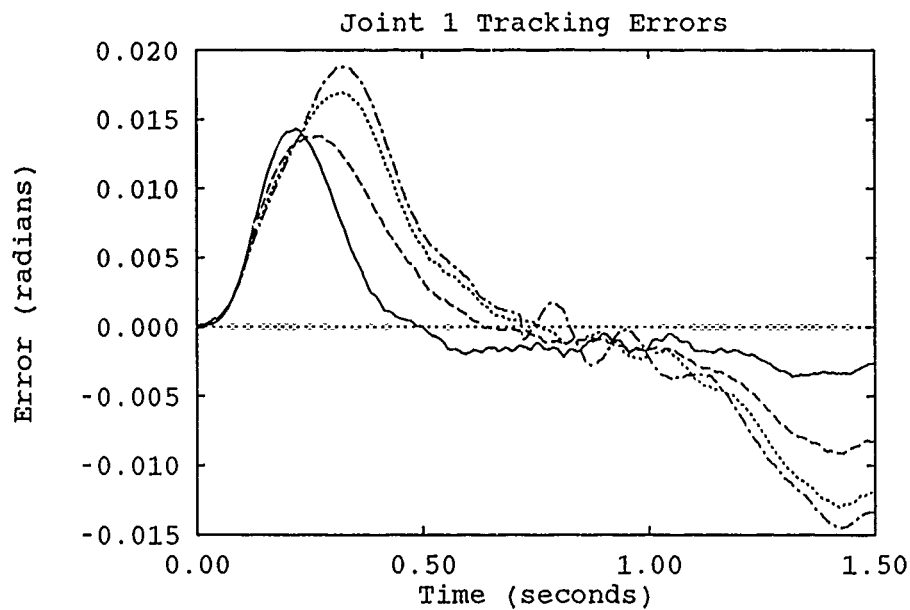


Figure A.16. 13-Parameter Uninitialized Adaptation Runs - Trajectory 1

—	First Run	Sixth Run
- - -	Third Run	- . - . -	Ninth Run

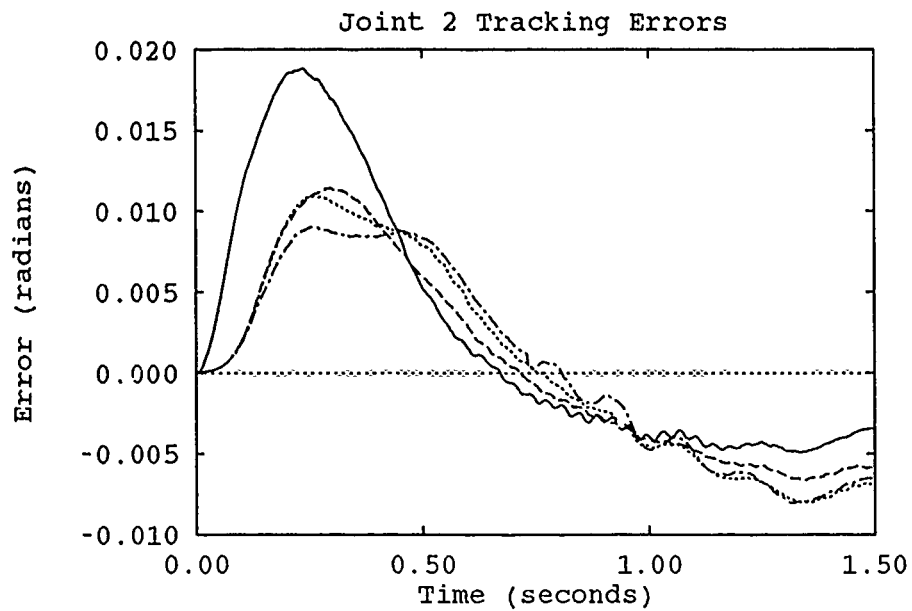


Figure A.17. 13-Parameter Uninitialized Adaptation Runs - Trajectory 1

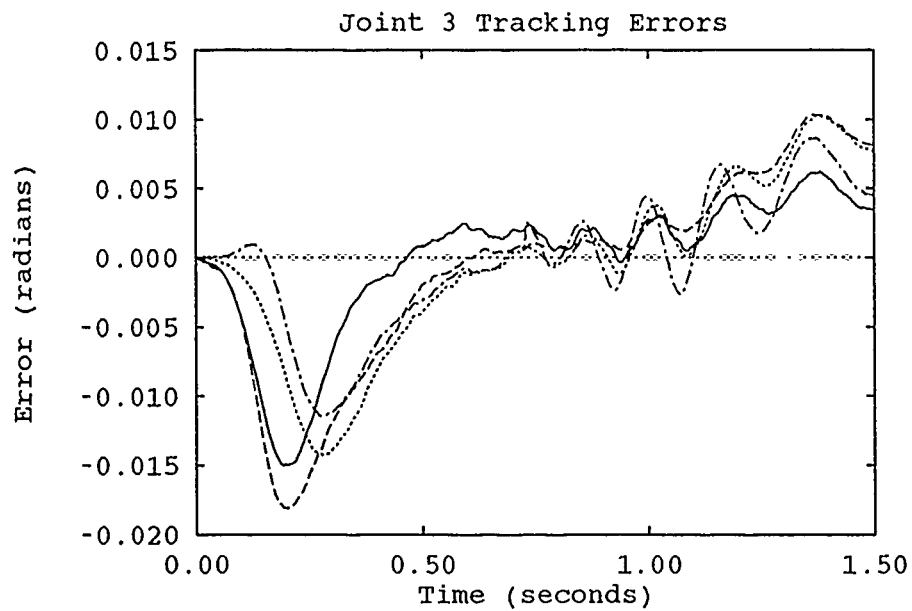


Figure A.18. 13-Parameter Uninitialized Adaptation Runs - Trajectory 1

—	First Run	Sixth Run
- - - -	Third Run	- · - · -	Ninth Run

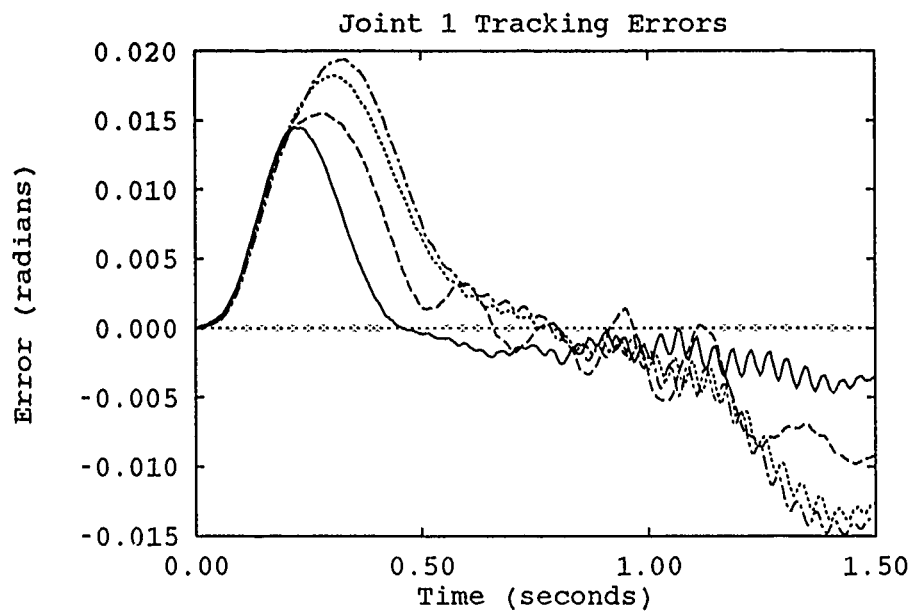


Figure A.19. 13-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

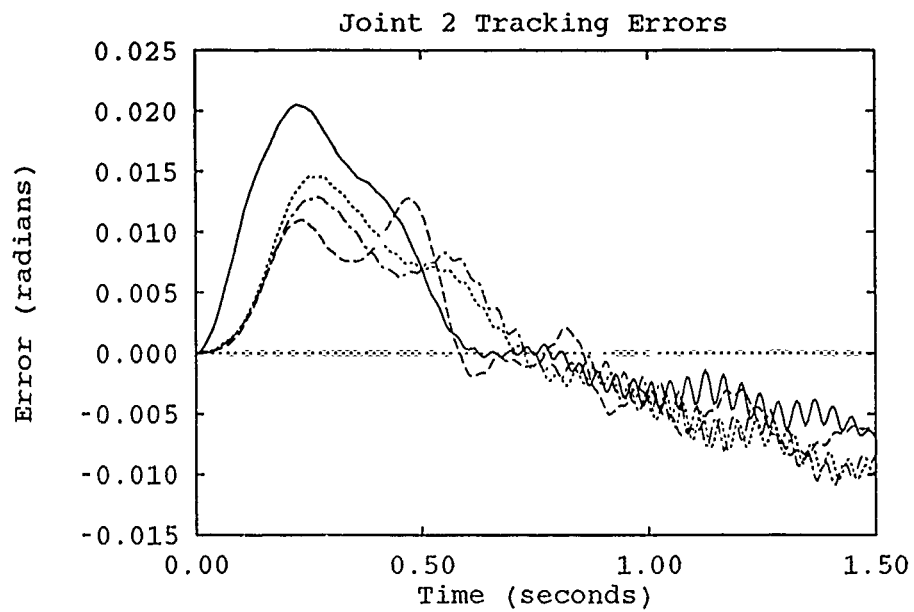


Figure A.20. 13-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

—	First Run	Sixth Run
----	Third Run	- - - -	Ninth Run

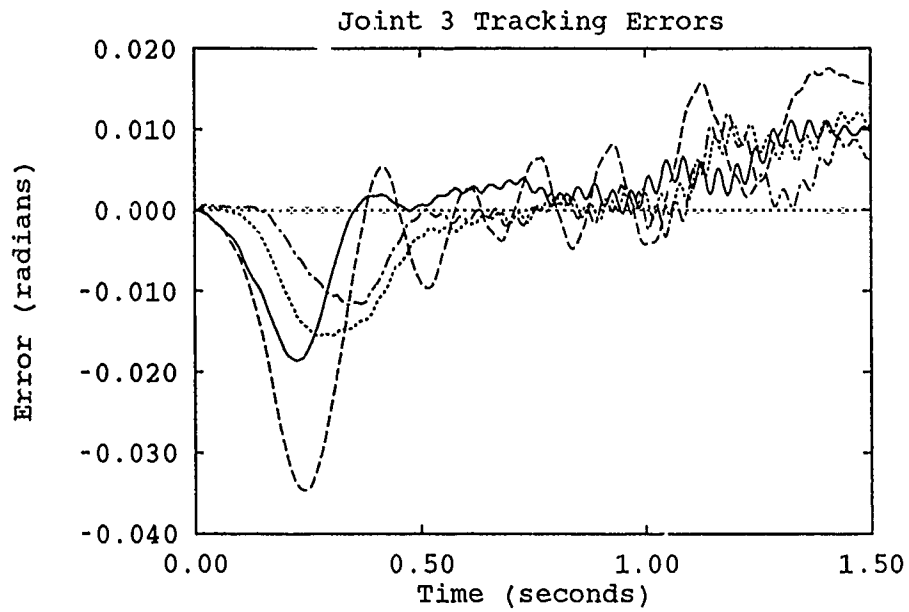


Figure A.21. 13-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

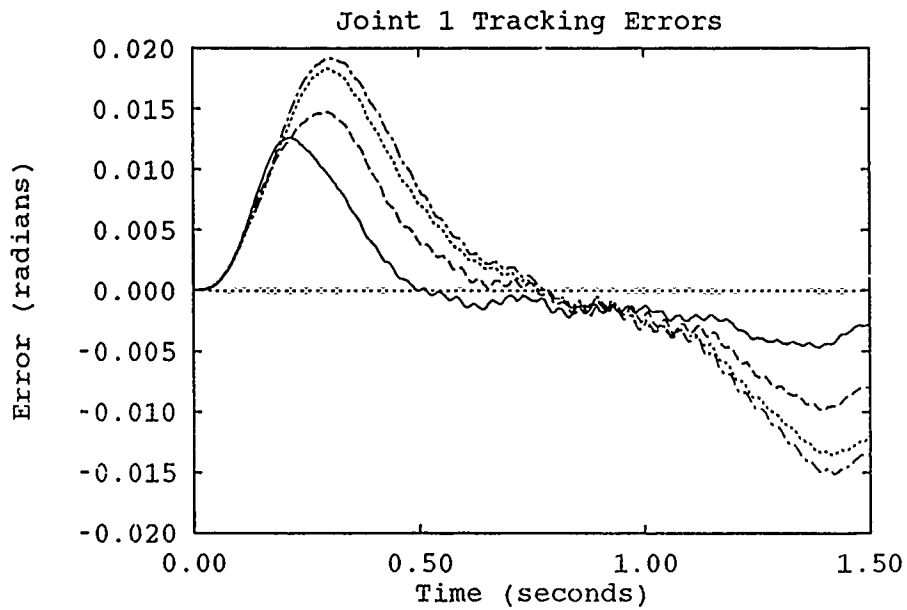


Figure A.22. 13-Parameter Uninitialized Adaptation Runs - Trajectory 6

—	First Run	Sixth Run
- - -	Third Run	- . - . -	Ninth Run

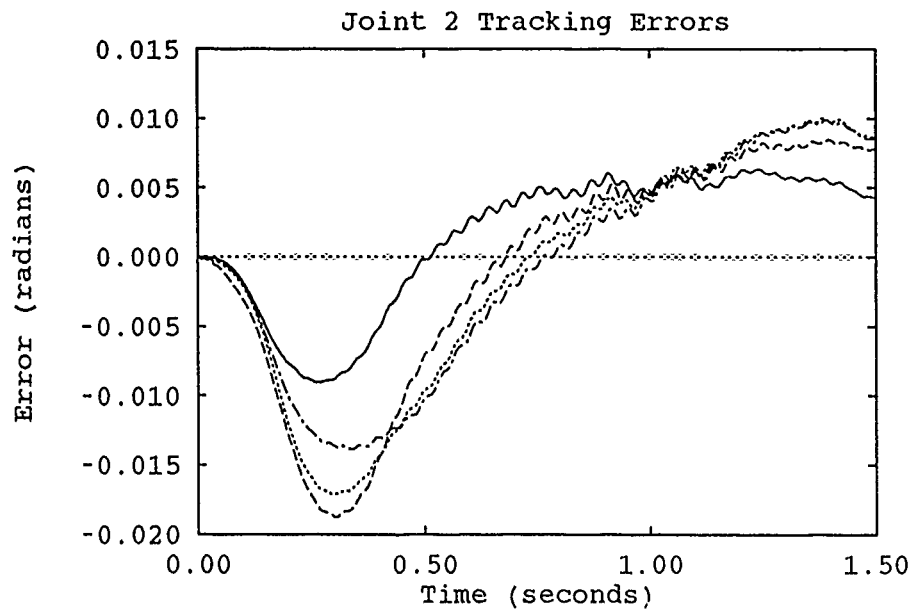


Figure A.23. 13-Parameter Uninitialized Adaptation Runs - Trajectory 6

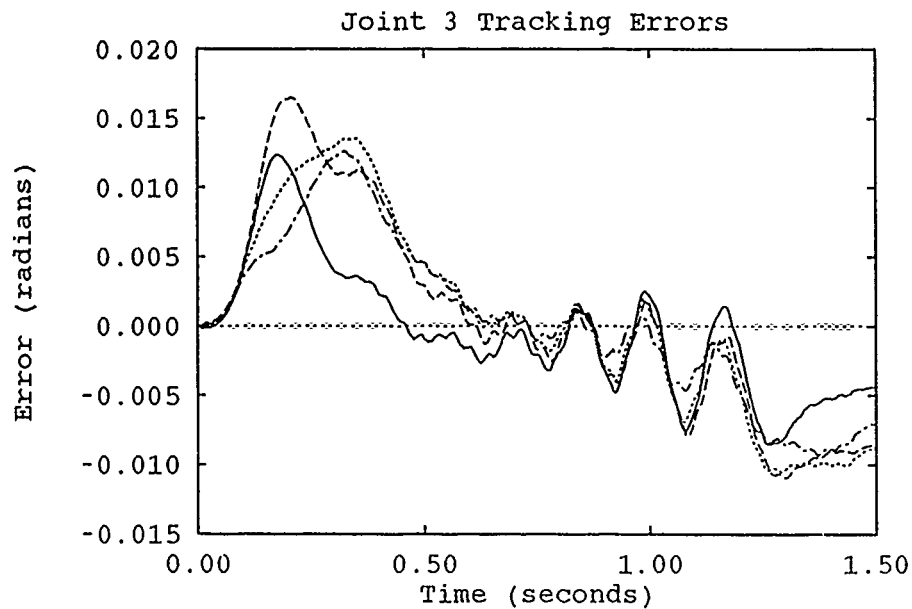


Figure A.24. 13-Parameter Uninitialized Adaptation Runs - Trajectory 6

—	First Run	Sixth Run
---	Third Run	- - - - -	Ninth Run

Appendix B. *16-Parameter Uninitialized Learning Runs*

This section contain the results of 16-Parameter adaptation runs for each test trajectory using uninitialized parameters. Each trajectory was run nine times to allow the controller to adapt.

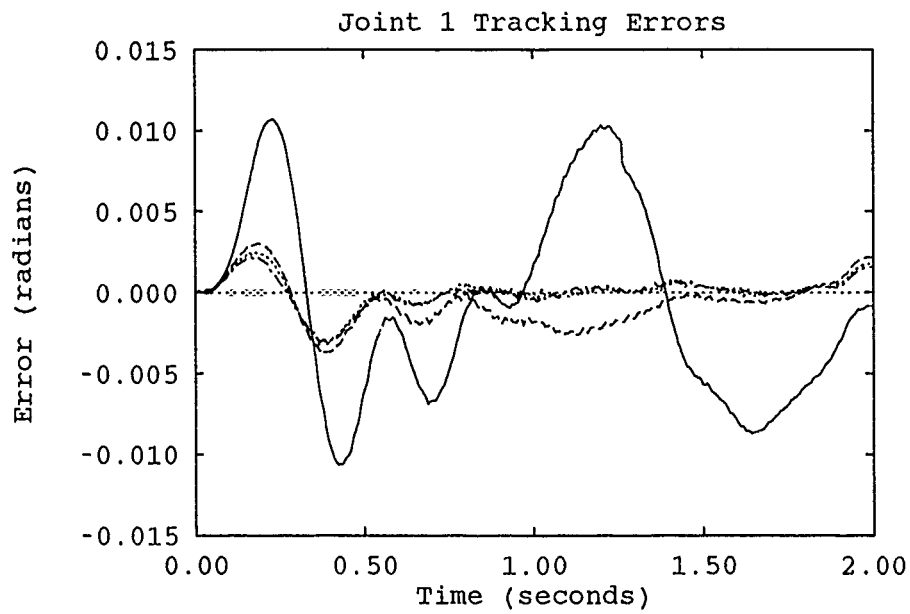


Figure B.1. 16-Parameter Uninitialized Adaptation Runs - Trajectory 2

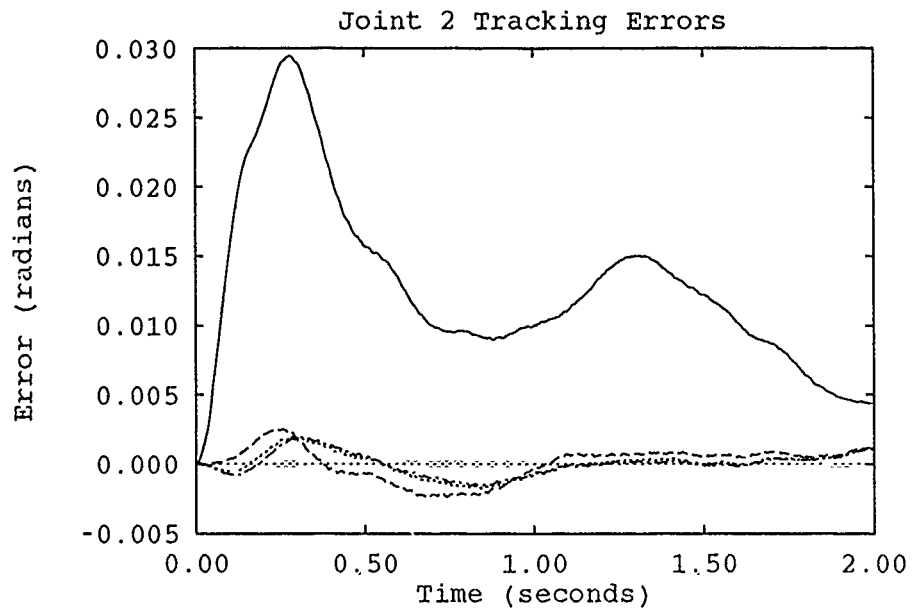


Figure B.2. 16-Parameter Uninitialized Adaptation Runs - Trajectory 2

—	First Run	Sixth Run
----	Third Run	- - - -	Ninth Run

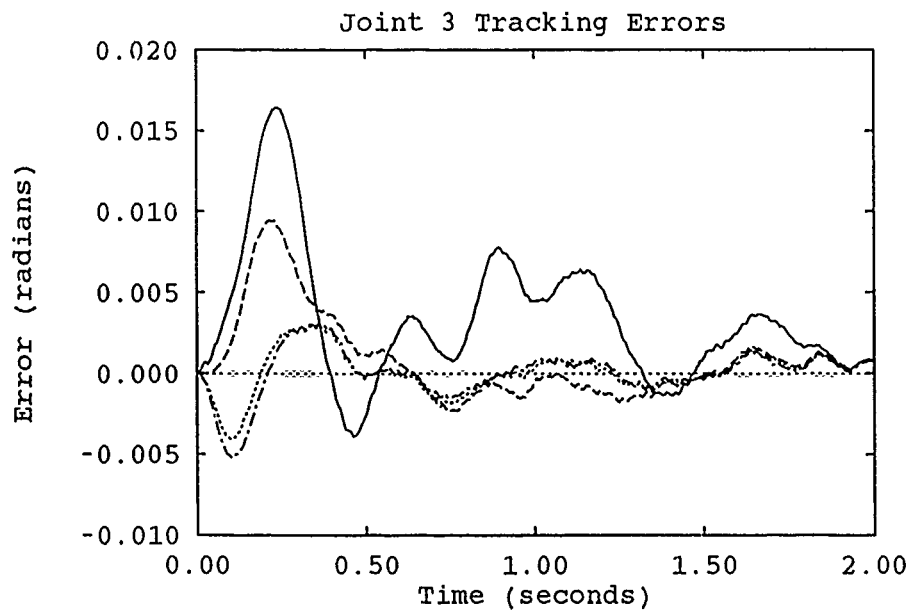


Figure B.3. 16-Parameter Uninitialized Adaptation Runs - Trajectory 2

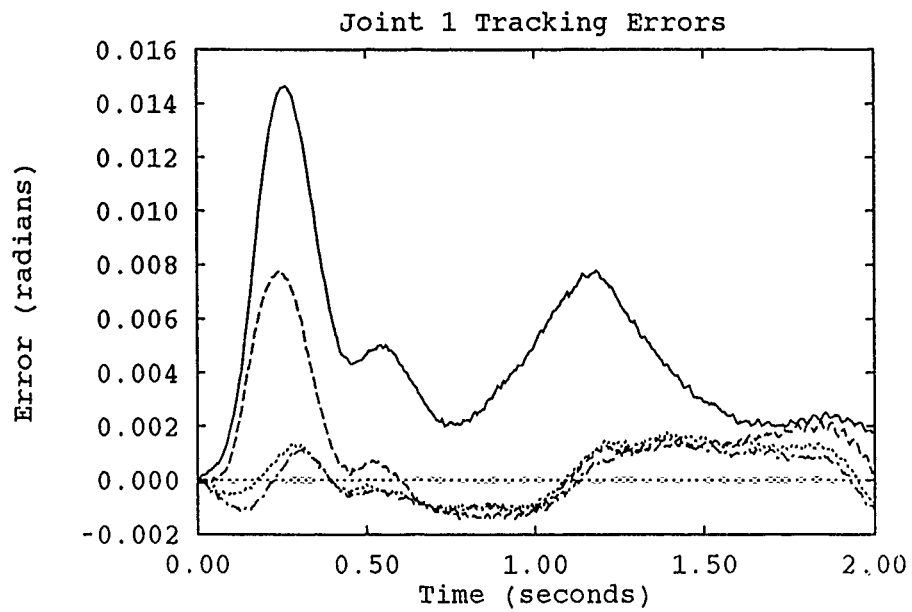


Figure B.4. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3

—	First Run	Sixth Run
- - -	Third Run	- . - . -	Ninth Run

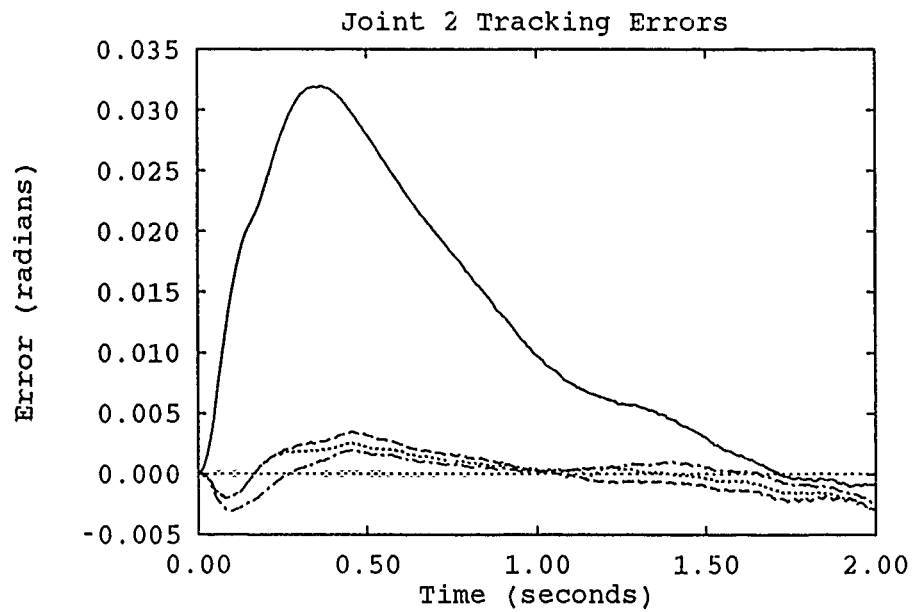


Figure B.5. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3

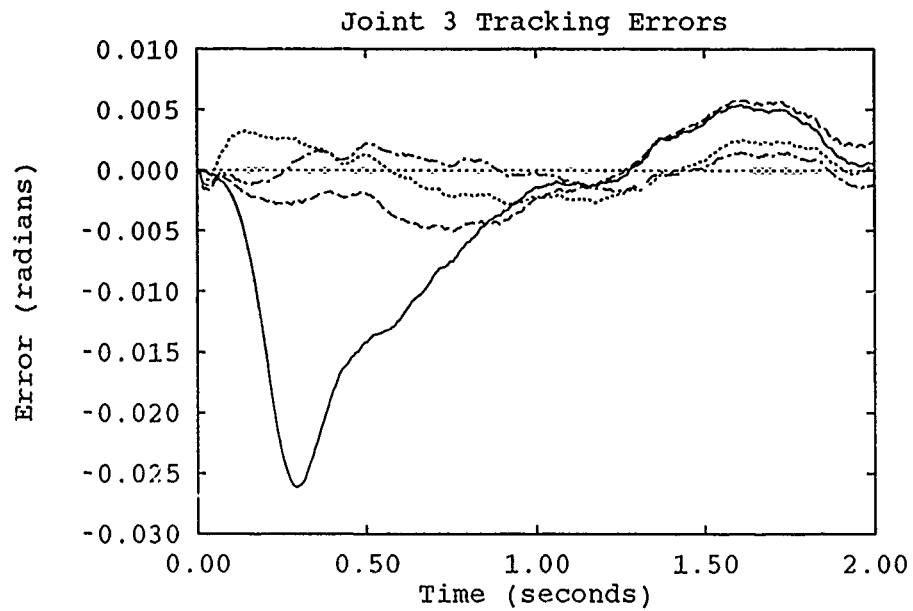


Figure B.6. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3

—	First Run	Sixth Run
---	Third Run	- - - -	Ninth Run

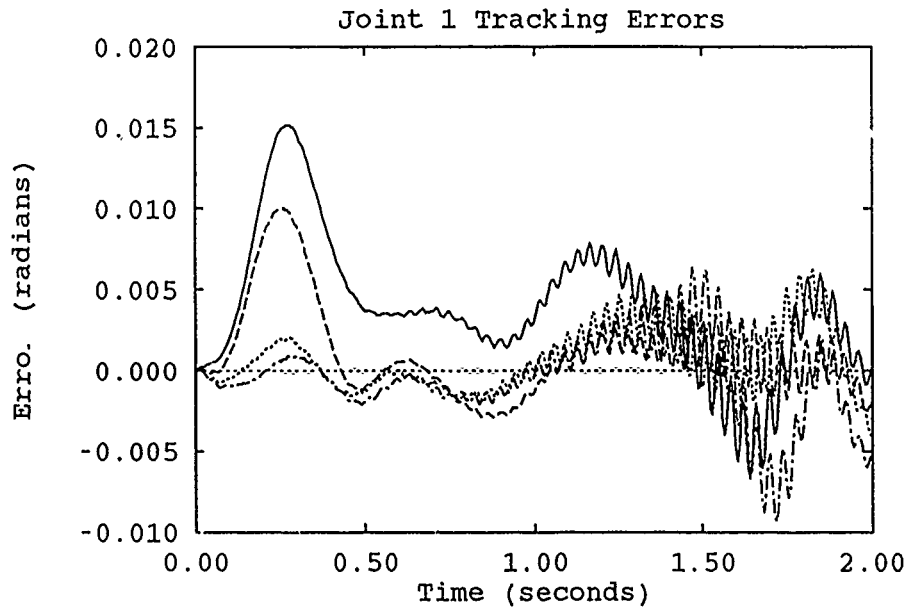


Figure B.7. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

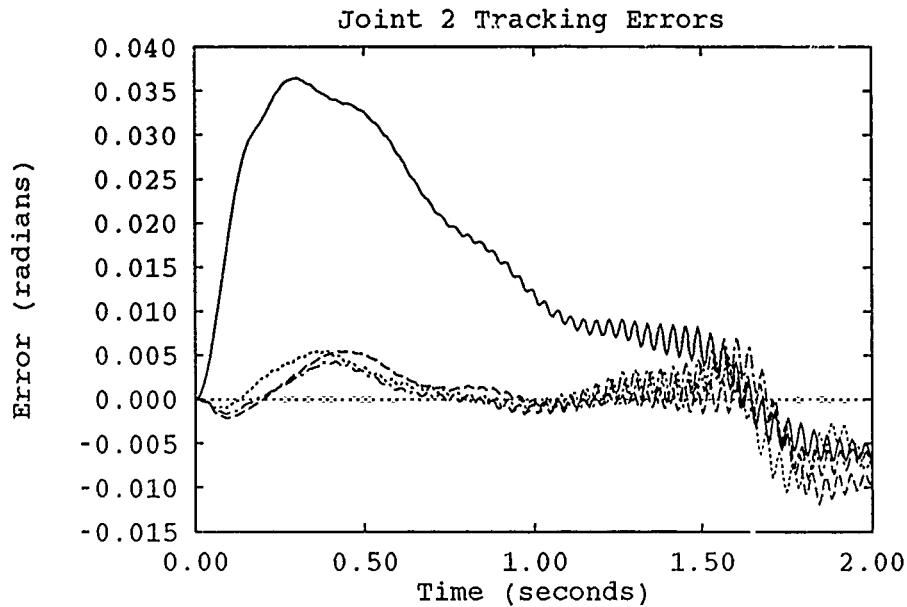


Figure B.8. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

—	First Run	Sixth Run
---	Third Run	-.-.-	Ninth Run

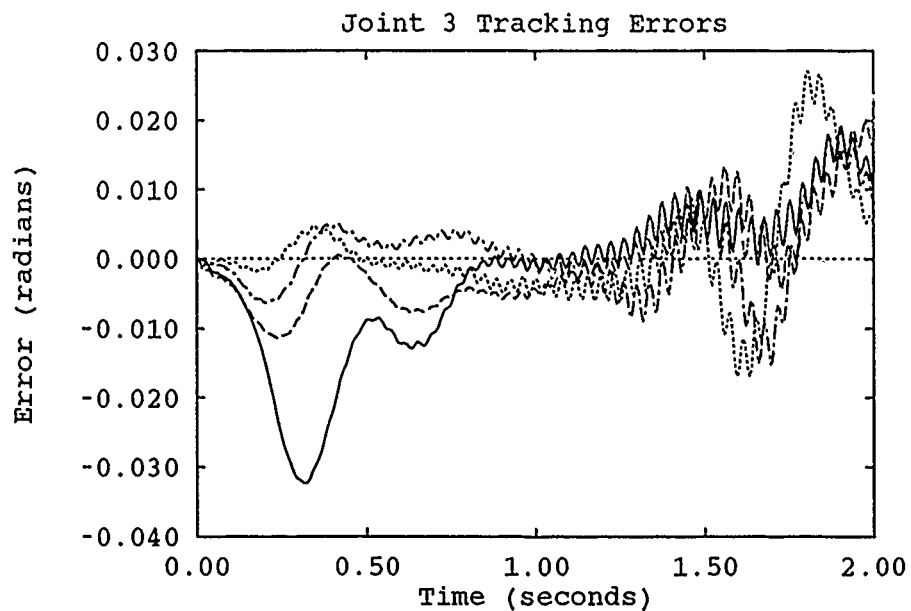


Figure B.9. 16-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

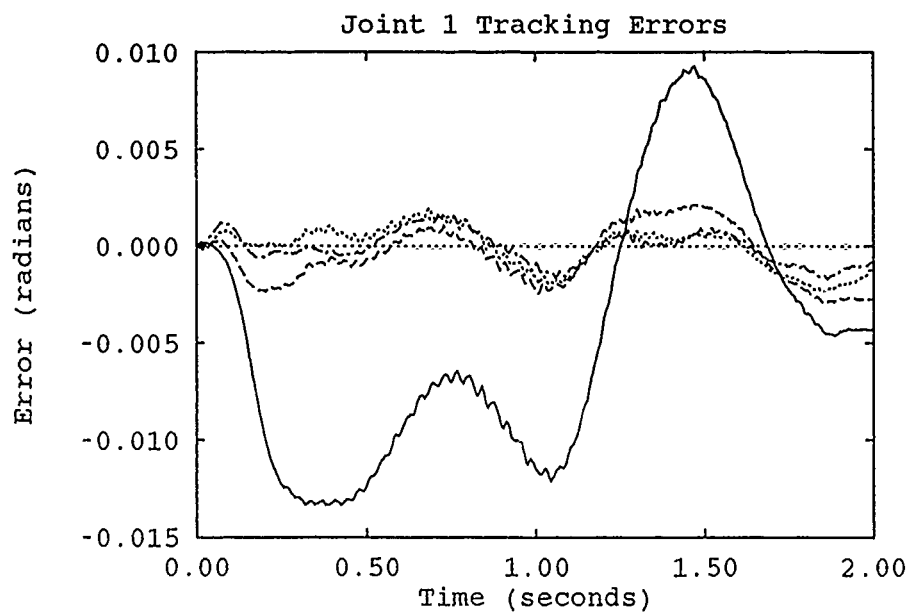


Figure B.10. 16-Parameter Uninitialized Adaptation Runs - Trajectory 4

—	First Run	Sixth Run
---	Third Run	-.-.-.-	Ninth Run

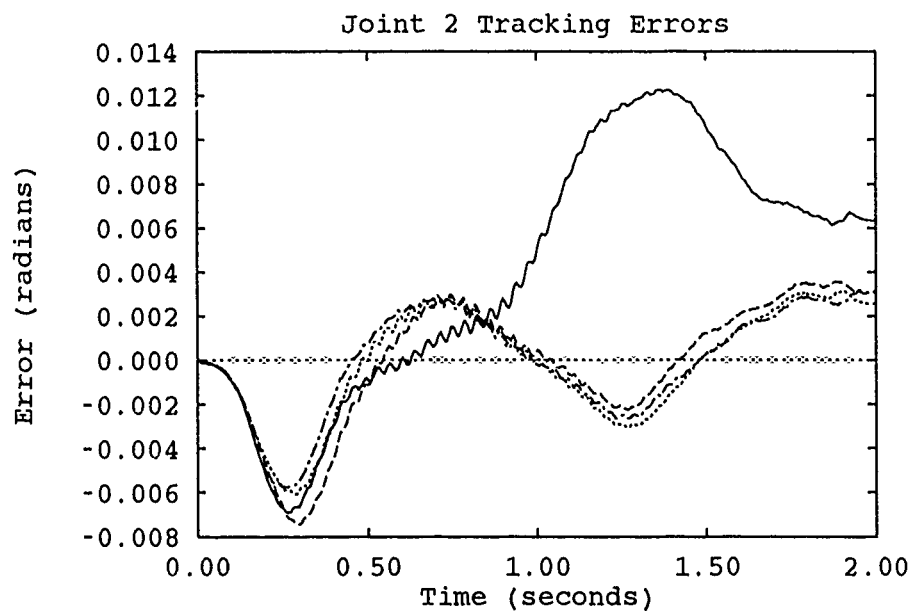


Figure B.11. 16-Parameter Uninitialized Adaptation Runs - Trajectory 4

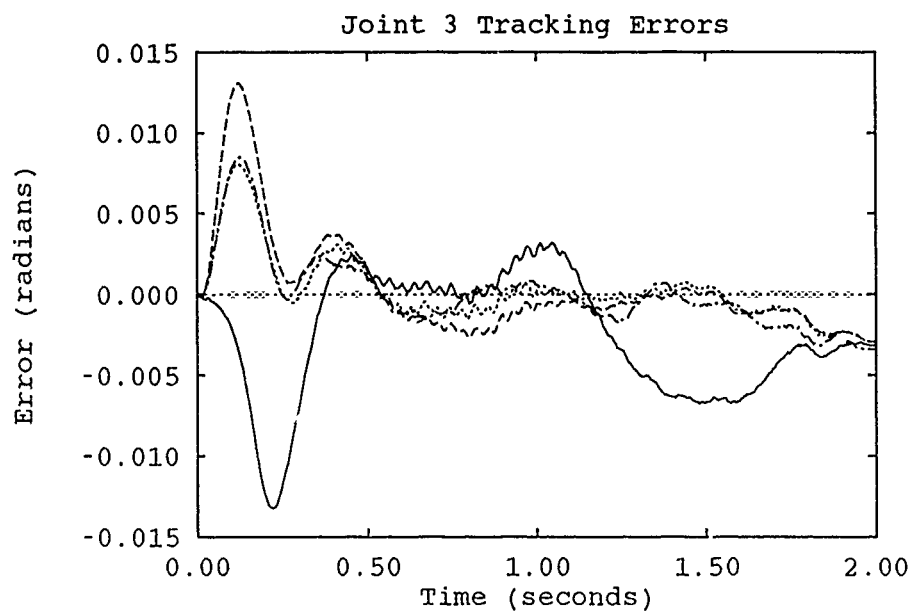


Figure B.12. 16-Parameter Uninitialized Adaptation Runs - Trajectory 4

—	First Run	Sixth Run
- - - -	Third Run	- . - . -	Ninth Run

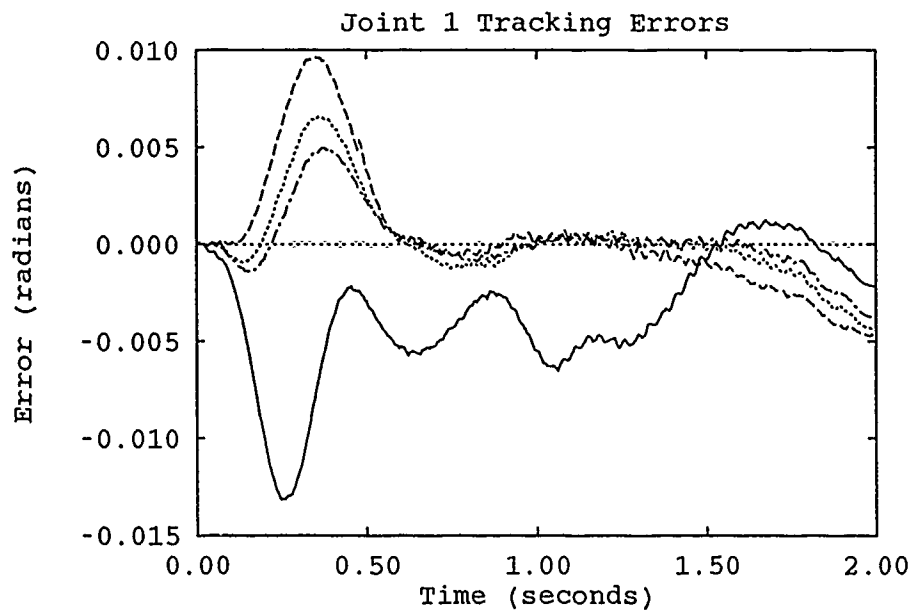


Figure B.13. 16-Parameter Uninitialized Adaptation Runs - Trajectory 5

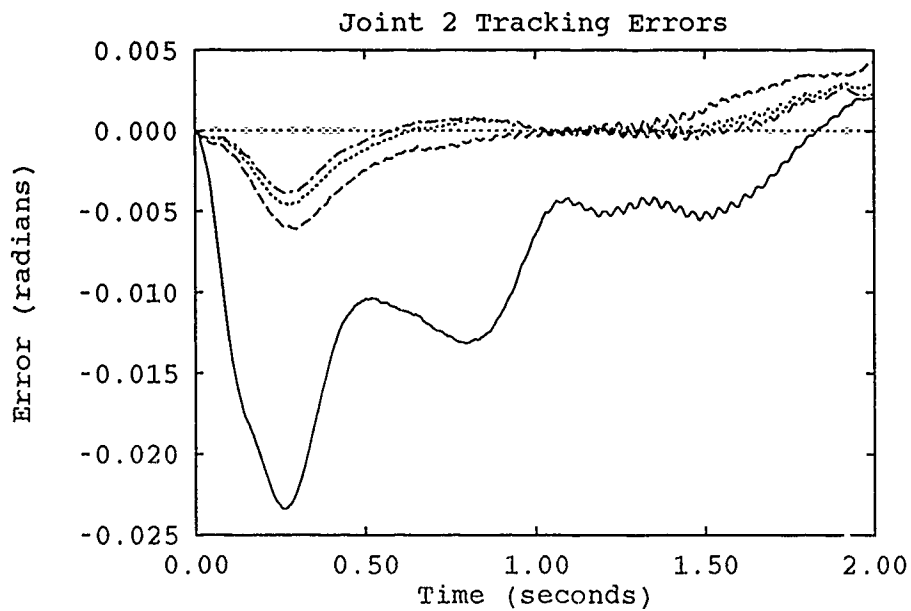


Figure B.14. 16-Parameter Uninitialized Adaptation Runs - Trajectory 5

—	First Run	Sixth Run
- - -	Third Run	- . - . -	Ninth Run

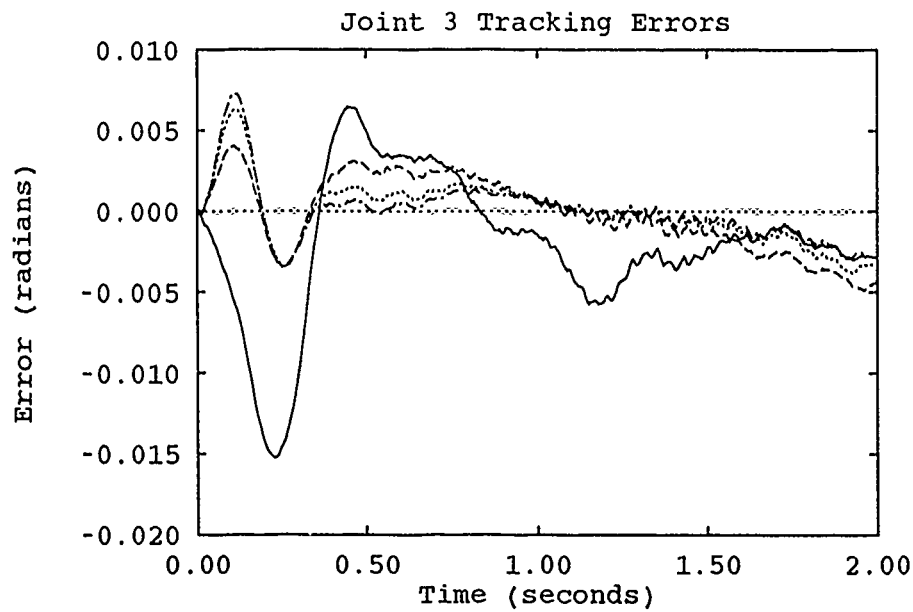


Figure B.15. 16-Parameter Uninitialized Adaptation Runs - Trajectory 5

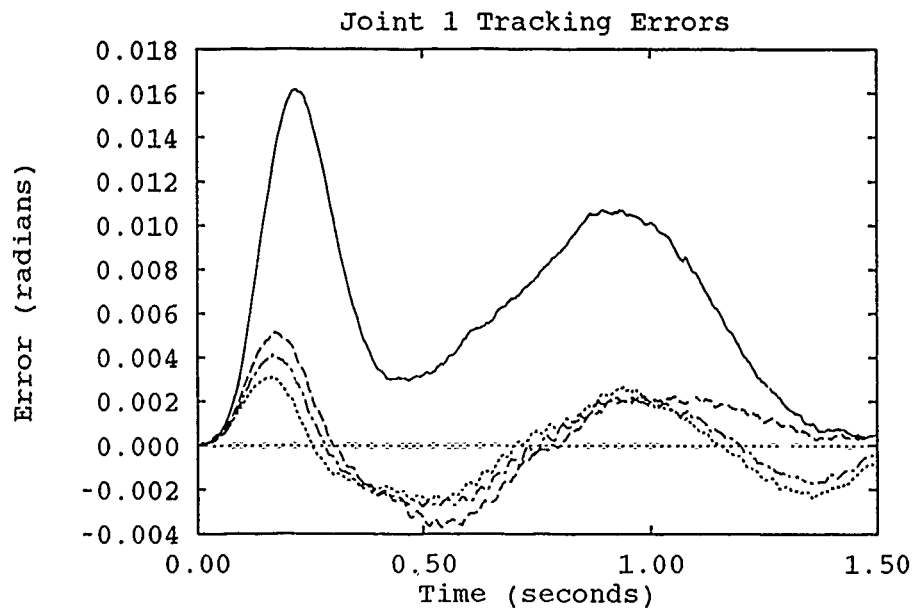


Figure B.16. 16-Parameter Uninitialized Adaptation Runs - Trajectory 1

—	First Run	Sixth Run
---	Third Run	-.-.-.-	Ninth Run

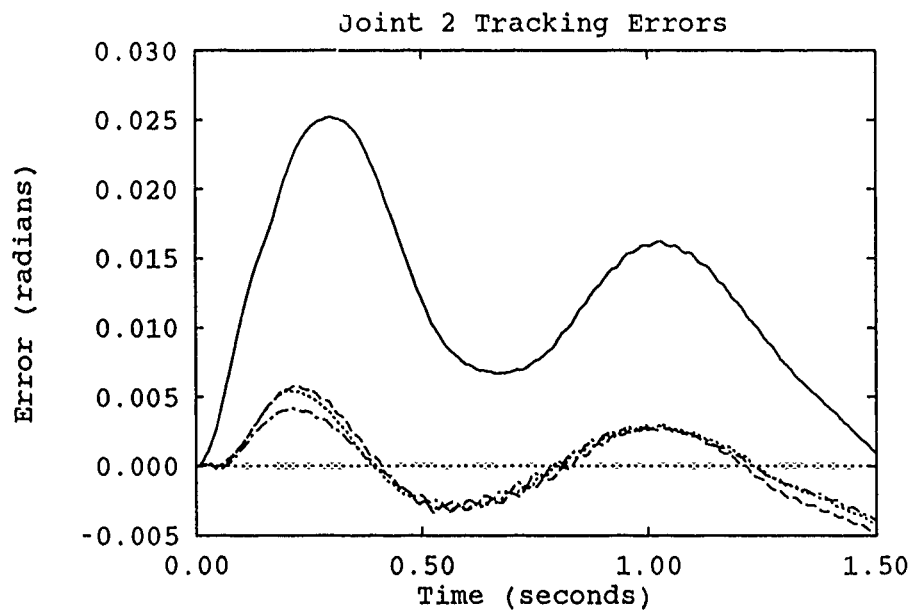


Figure B.17. 16-Parameter Uninitialized Adaptation Runs - Trajectory 1

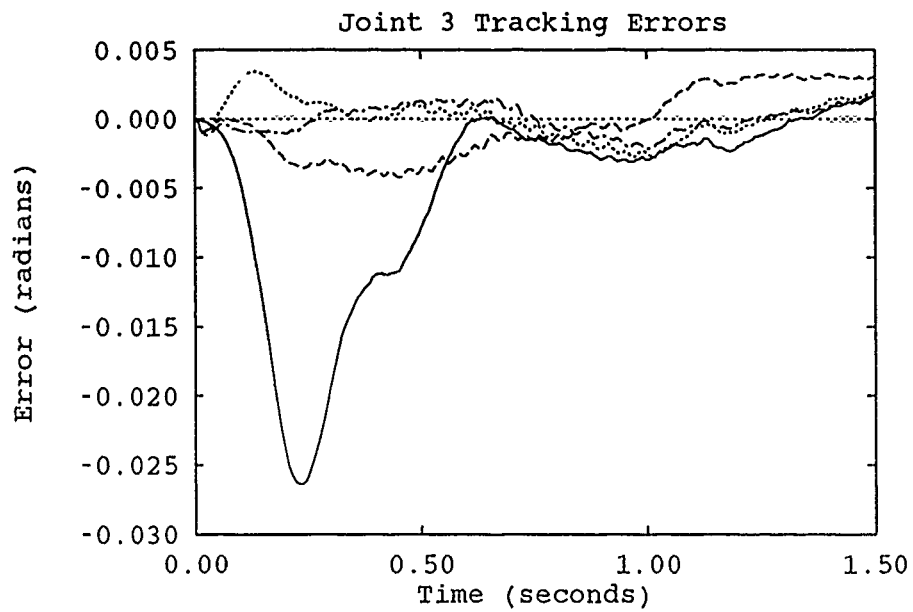


Figure B.18. 16-Parameter Uninitialized Adaptation Runs - Trajectory 1

—	First Run	Sixth Run
---	Third Run	-.-.-	Ninth Run

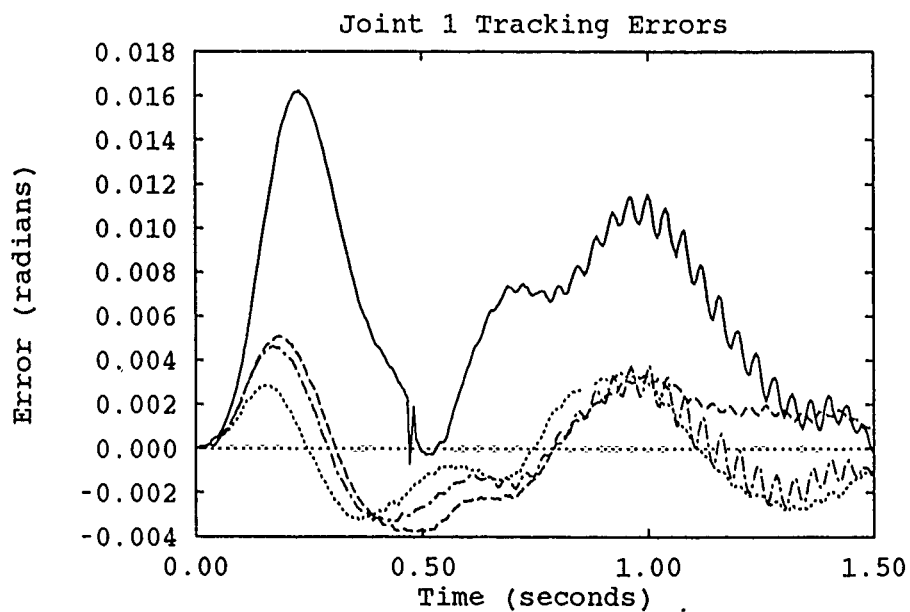


Figure B.19. 16-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

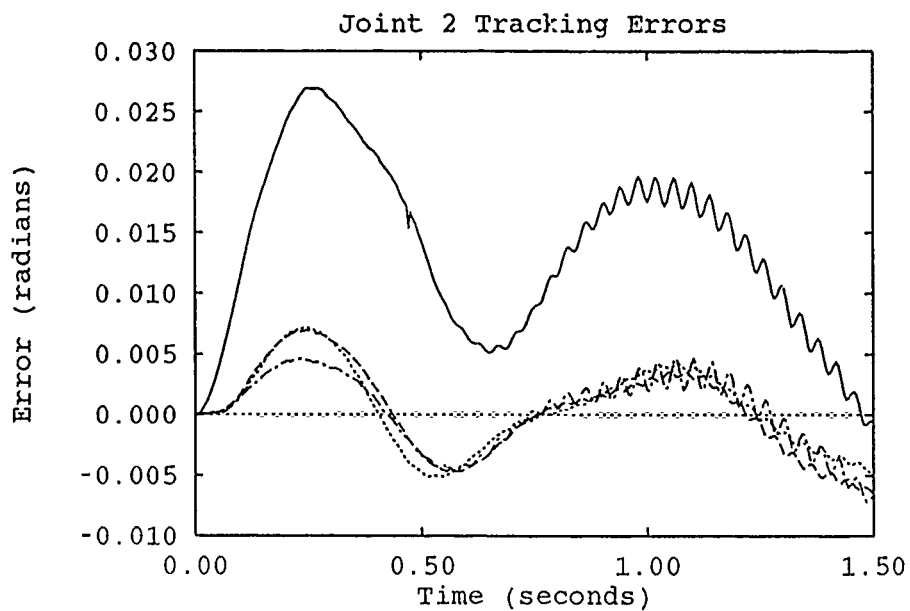


Figure B.20. 16-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

—	First Run	Sixth Run
---	Third Run	-.-.-.-	Ninth Run

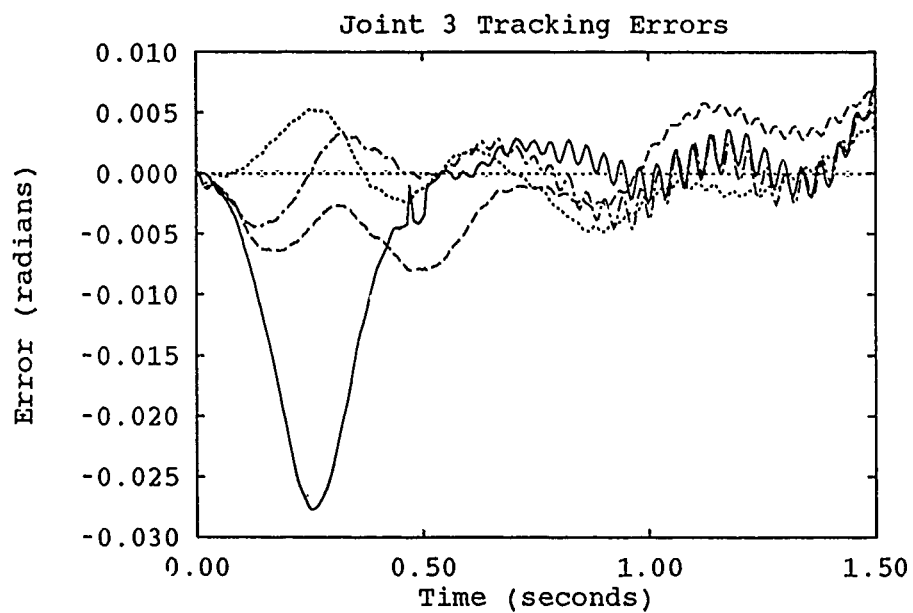


Figure B.21. 16-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

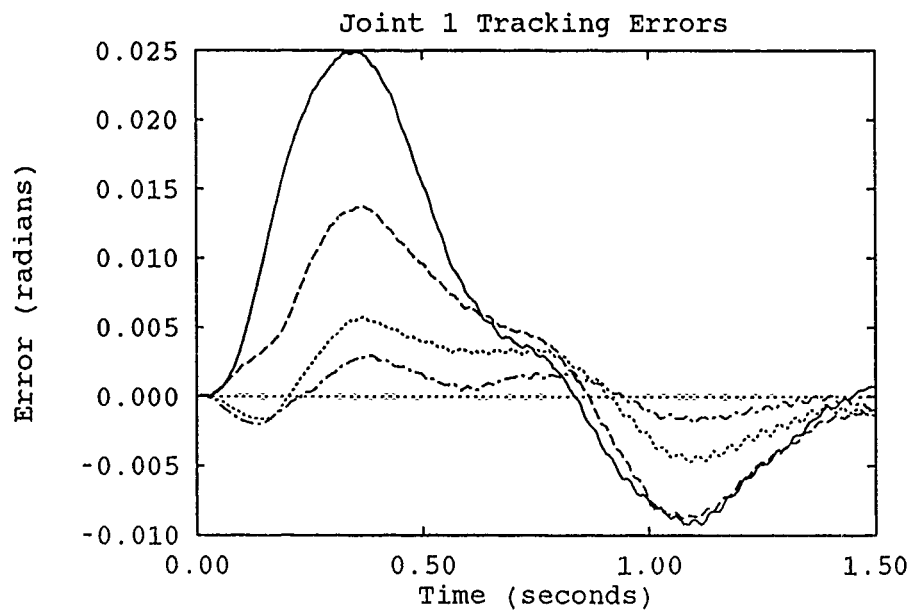


Figure B.22. 16-Parameter Uninitialized Adaptation Runs - Trajectory 6

—	First Run	Sixth Run
---	Third Run	-.-.-	Ninth Run

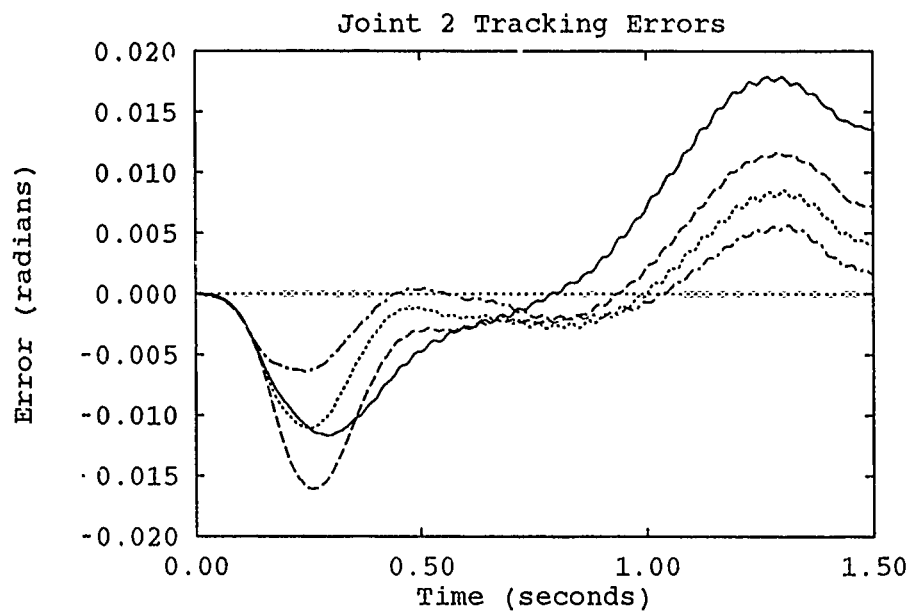


Figure B.23. 16-Parameter Uninitialized Adaptation Runs - Trajectory 6

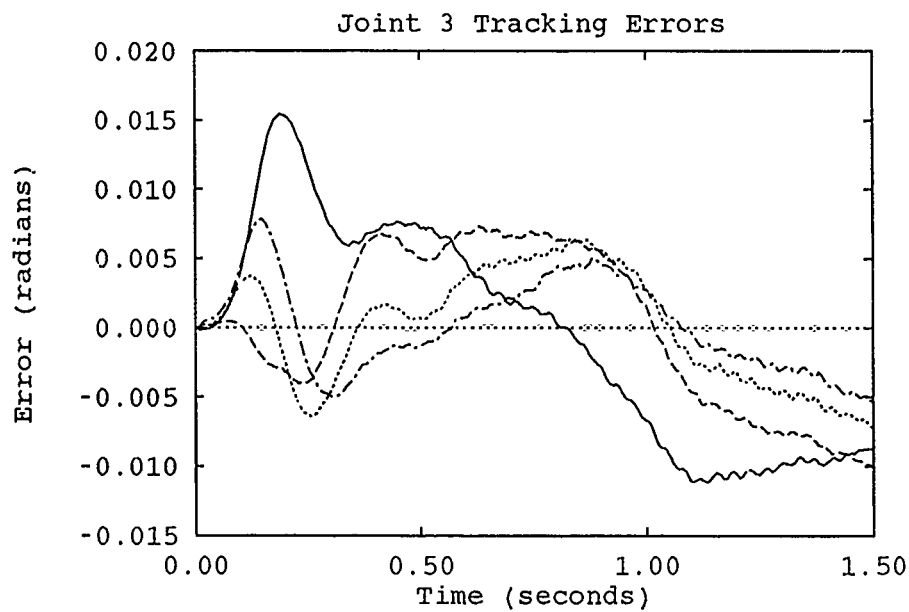


Figure B.24. 16-Parameter Uninitialized Adaptation Runs - Trajectory 6

—	First Run	Sixth Run
---	Third Run	-.-.-.-	Ninth Run

Appendix C. *19-Parameter Uninitialized Learning Runs*

This section contain the results of 19-parameter adaptation runs for each test trajectory using uninitialized parameters. Each trajectory was run nine times to allow the controller to adapt.

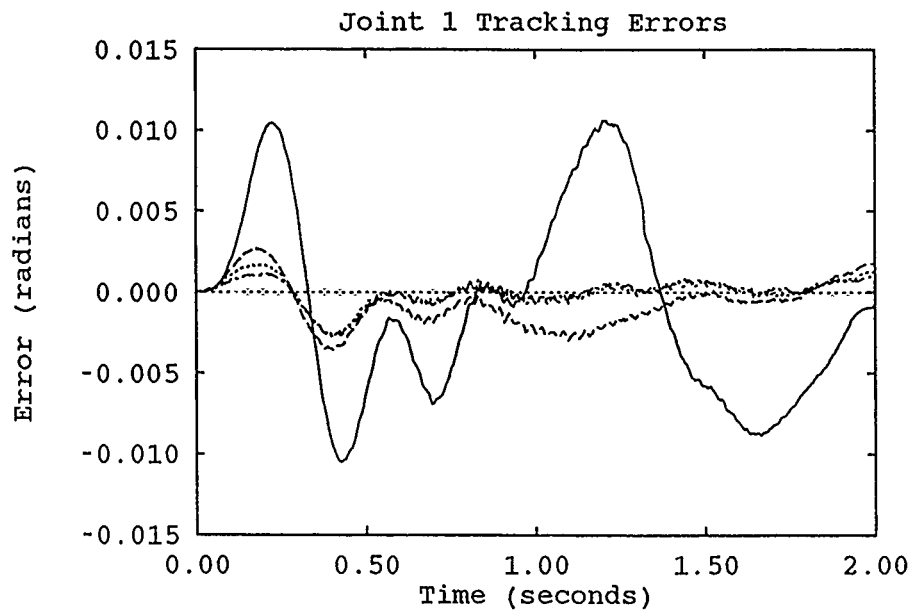


Figure C.1. 19-Parameter Uninitialized Adaptation Runs - Trajectory 2

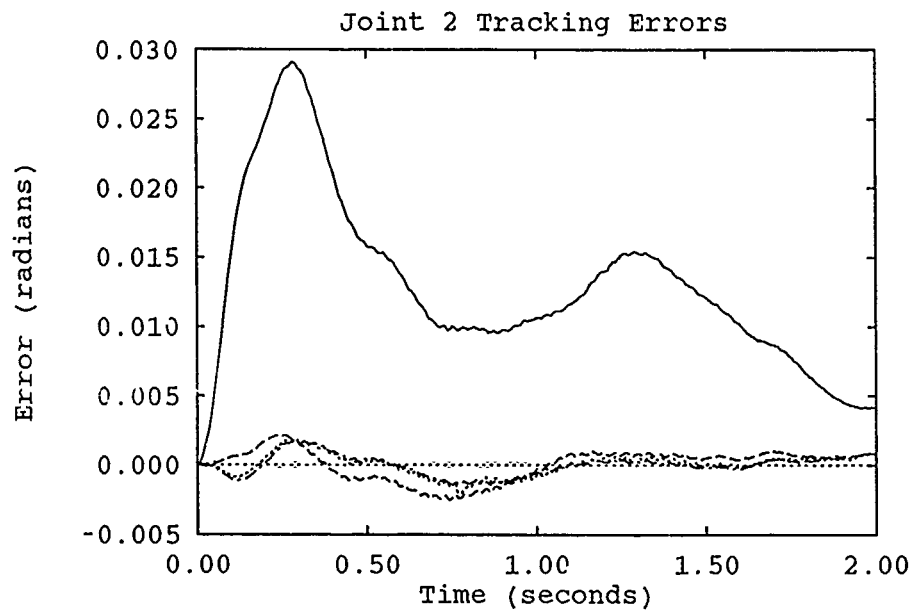


Figure C.2. 19-Parameter Uninitialized Adaptation Runs - Trajectory 2

—	First Run	Sixth Run
---	Third Run	-.-.-.-	Ninth Run

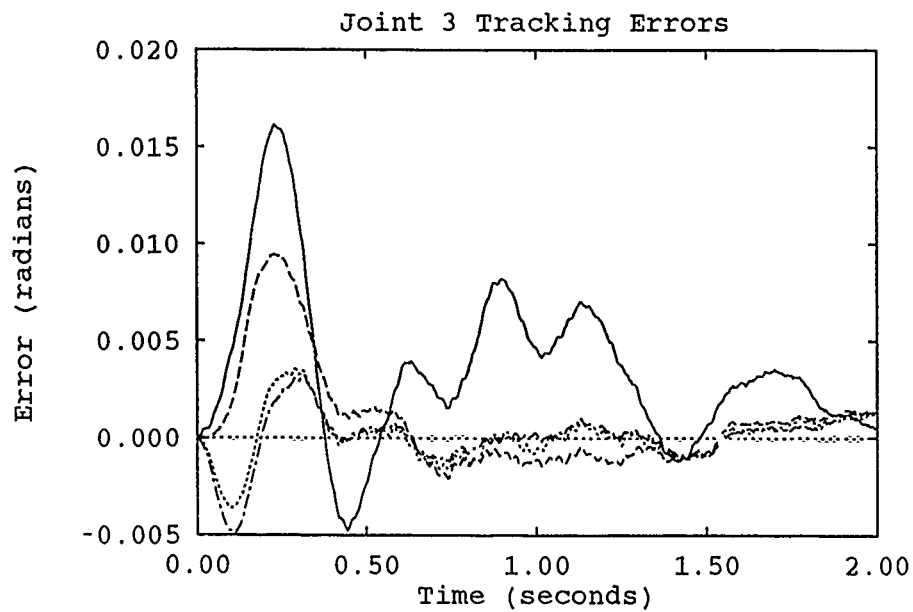


Figure C.3. 19-Parameter Uninitialized Adaptation Runs - Trajectory 2

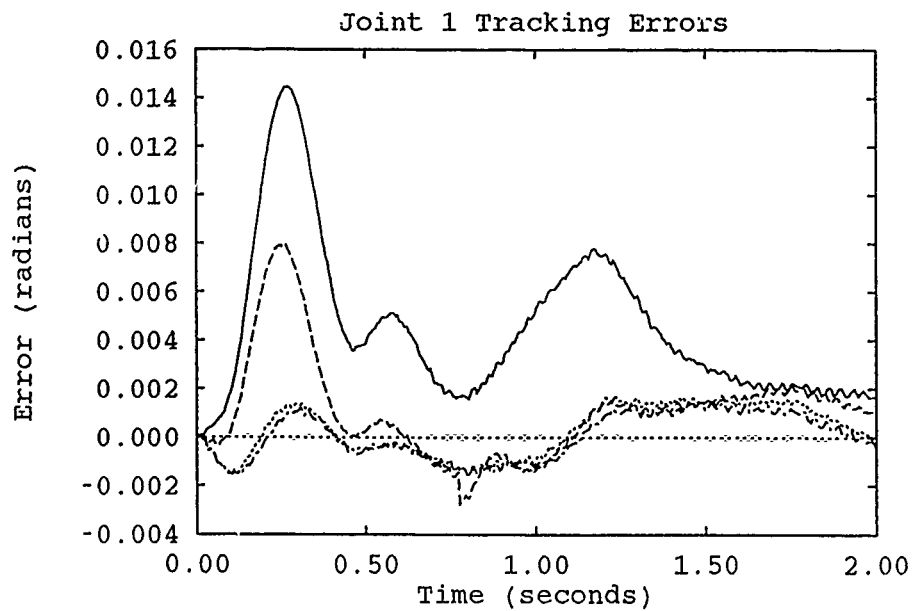


Figure C.4. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3

—	First Run	Sixth Run
- - -	Third Run	- . - .	Ninth Run

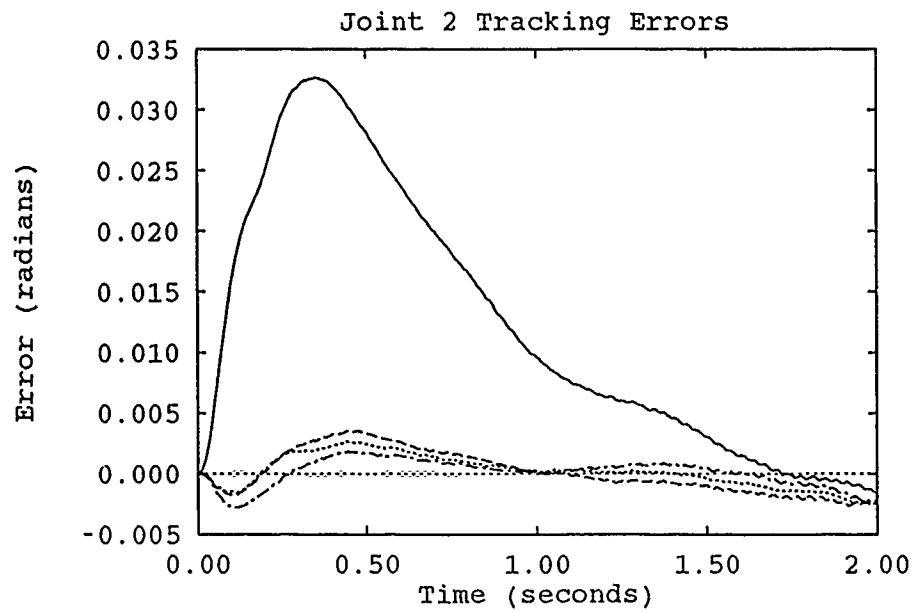


Figure C.5. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3

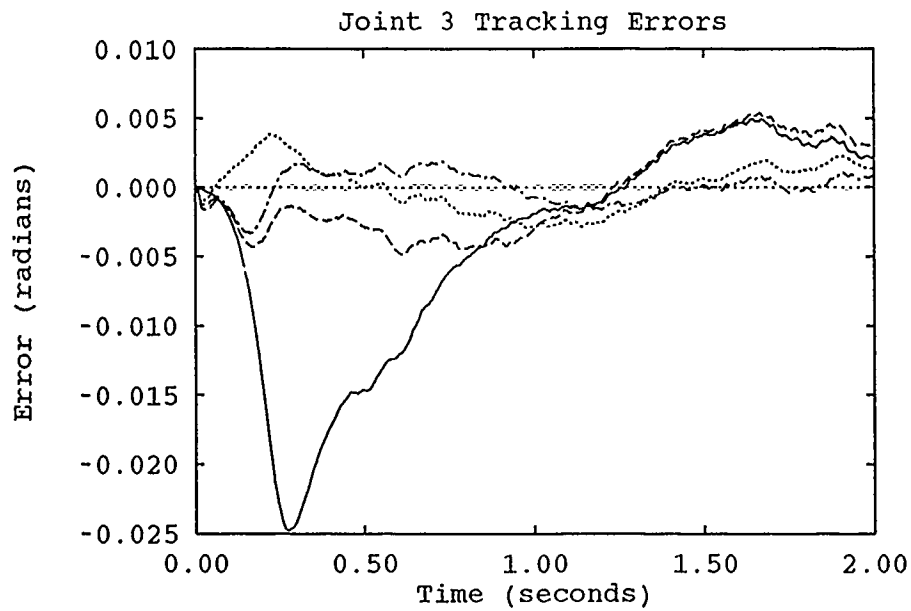


Figure C.6. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3

—	First Run	Sixth Run
- - - -	Third Run	- . - . -	Ninth Run

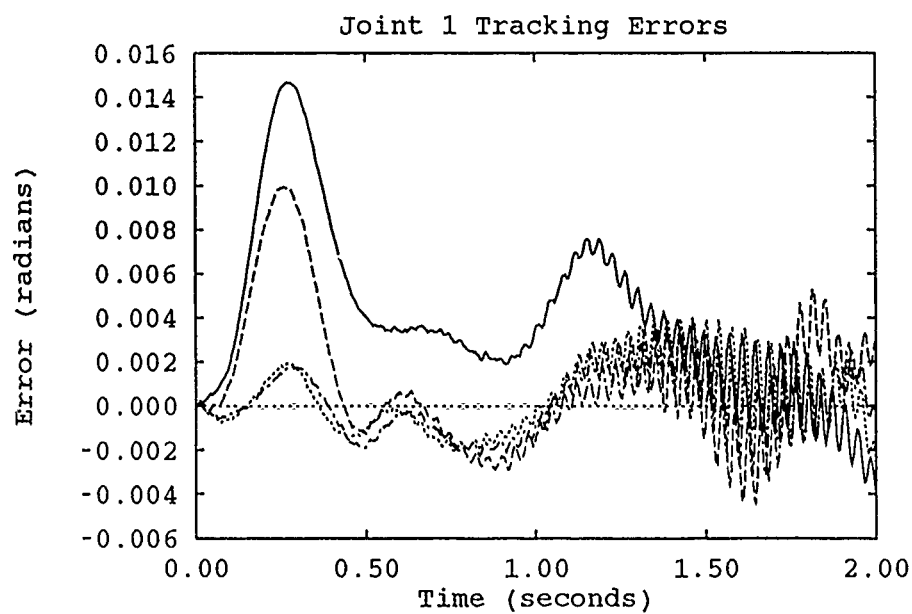


Figure C.7. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

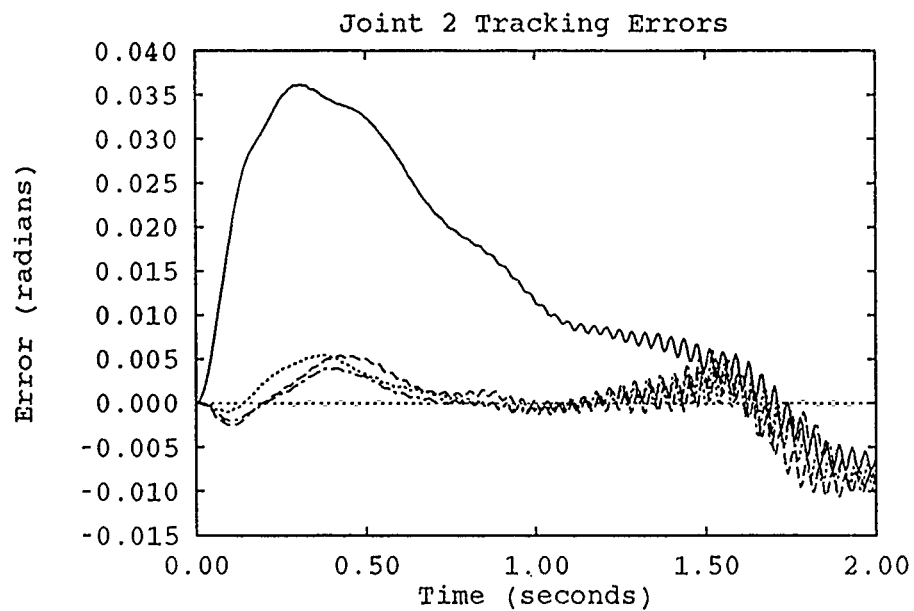


Figure C.8. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

————	First Run	Sixth Run
-----	Third Run	- - - -	Ninth Run

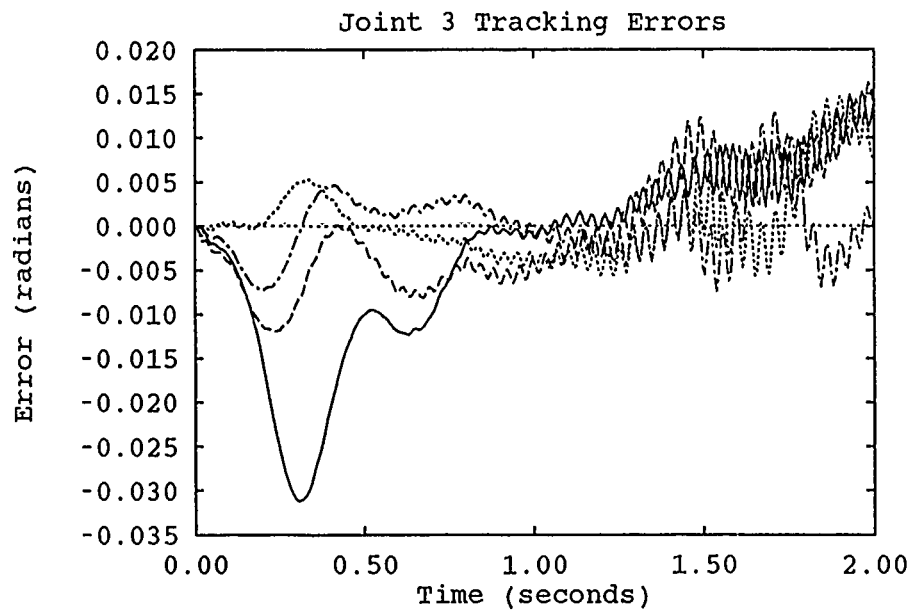


Figure C.9. 19-Parameter Uninitialized Adaptation Runs - Trajectory 3 w/ Payload

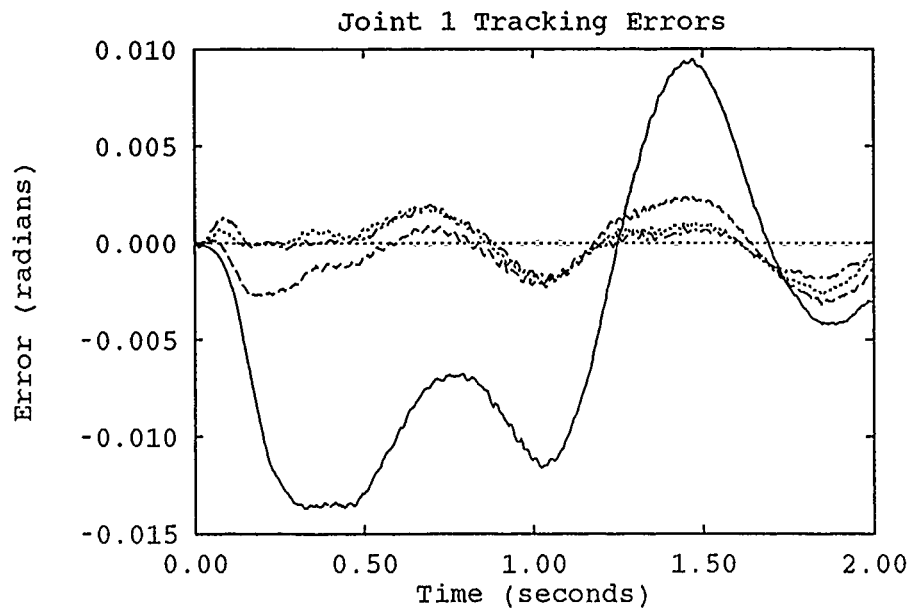


Figure C.10. 19-Parameter Uninitialized Adaptation Runs - Trajectory 4

—	First Run	Sixth Run
----	Third Run	- - - - -	Ninth Run

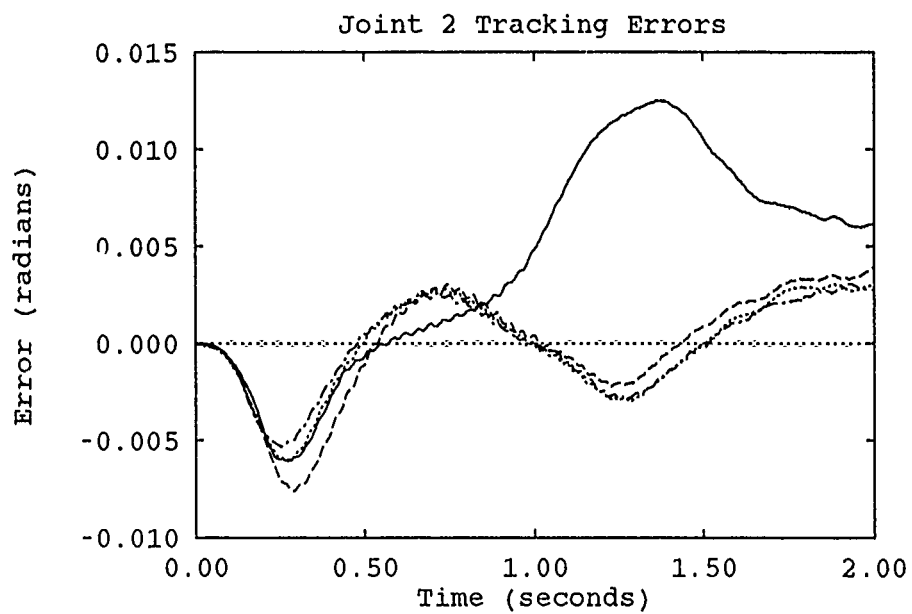


Figure C.11. 19-Parameter Uninitialized Adaptation Runs - Trajectory 4

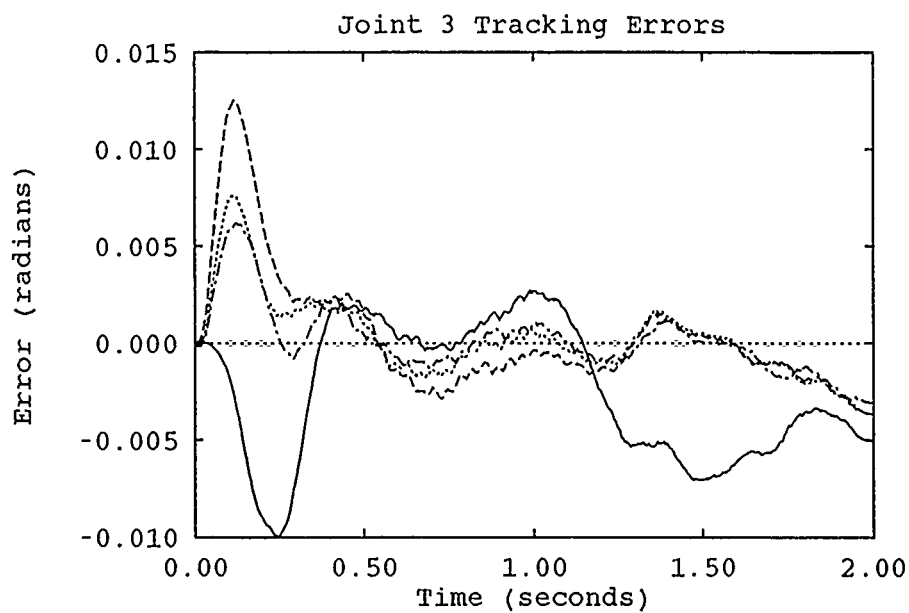


Figure C.12. 19-Parameter Uninitialized Adaptation Runs - Trajectory 4

—	First Run	Sixth Run
- - -	Third Run	- . - . -	Ninth Run

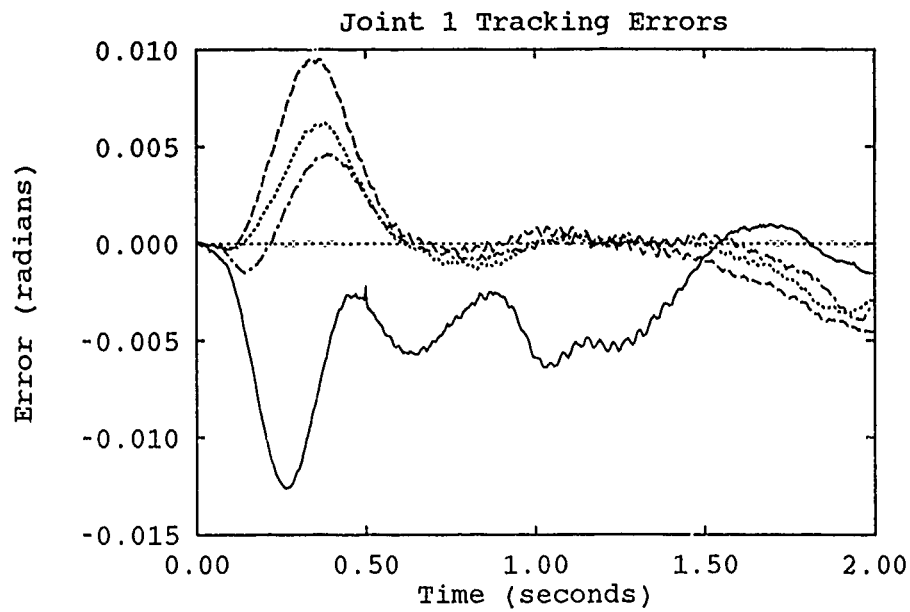


Figure C.13. 19-Parameter Uninitialized Adaptation Runs - Trajectory 5

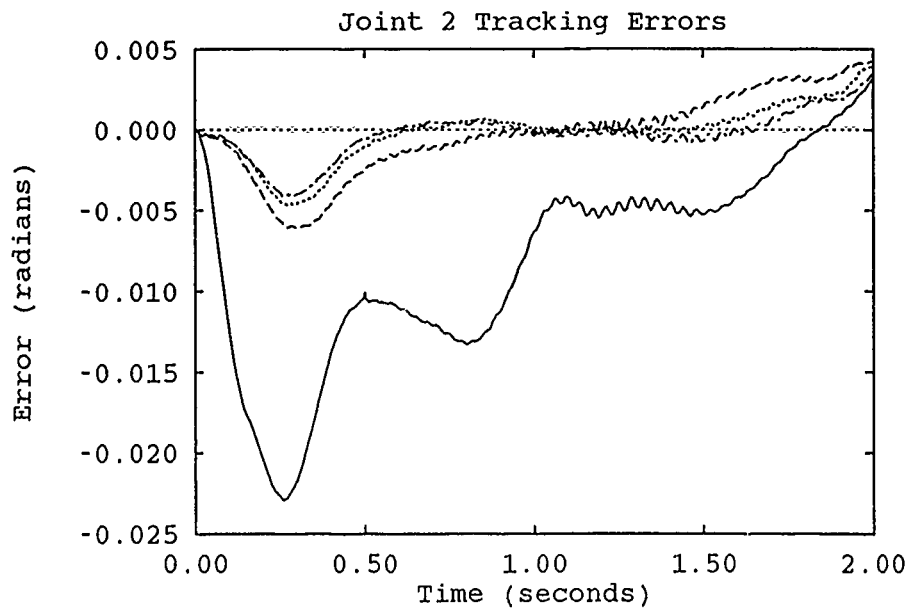


Figure C.14. 19-Parameter Uninitialized Adaptation Runs - Trajectory 5

—	First Run	·····	Sixth Run
---	Third Run	-·-·-	Ninth Run

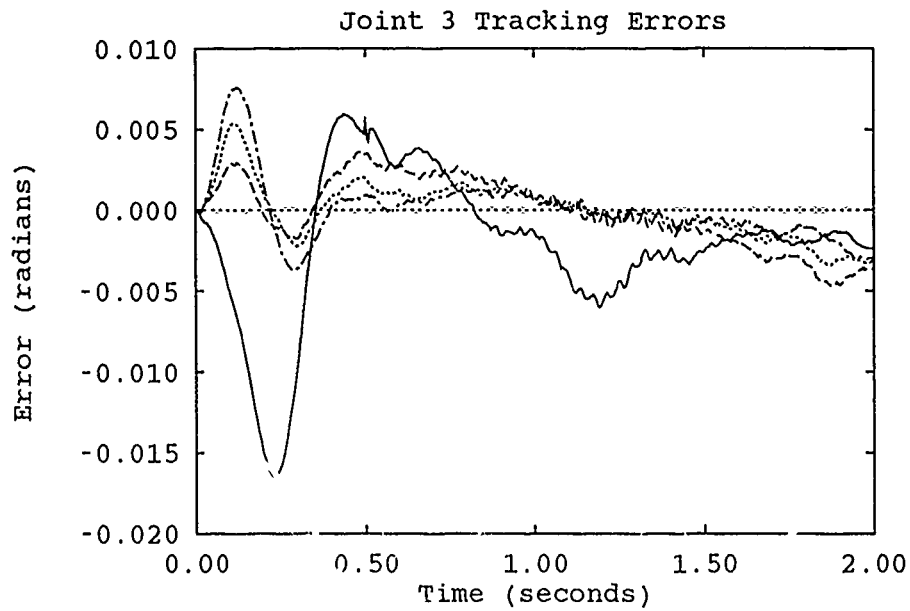


Figure C.15. 19-Parameter Uninitialized Adaptation Runs - Trajectory 5

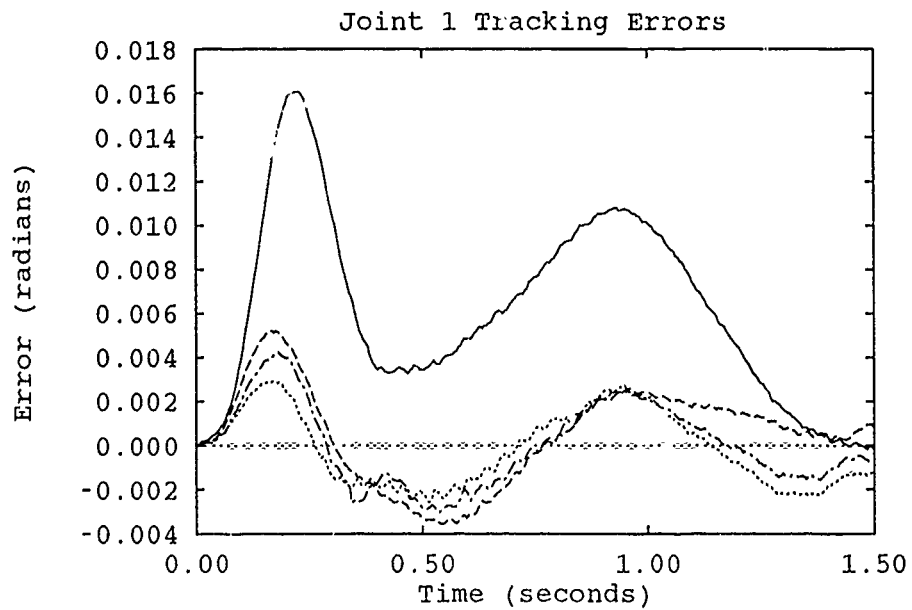


Figure C.16. 19 Parameter Uninitialized Adaptation Runs Trajectory 1

—	First Run	·····	Sixth Run
---	Third Run	- - -	Ninth Run

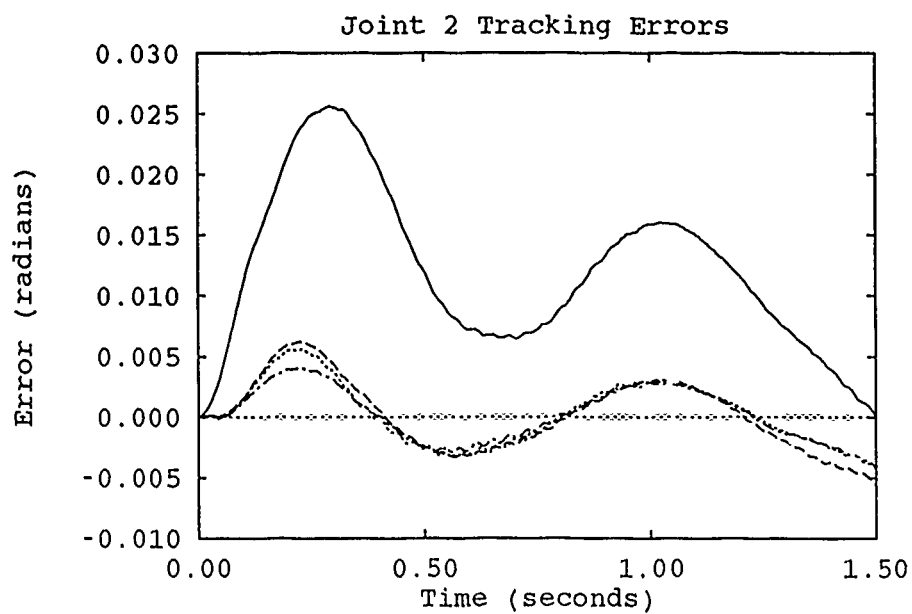


Figure C.17. 19-Parameter Uninitialized Adaptation Runs - Trajectory 1

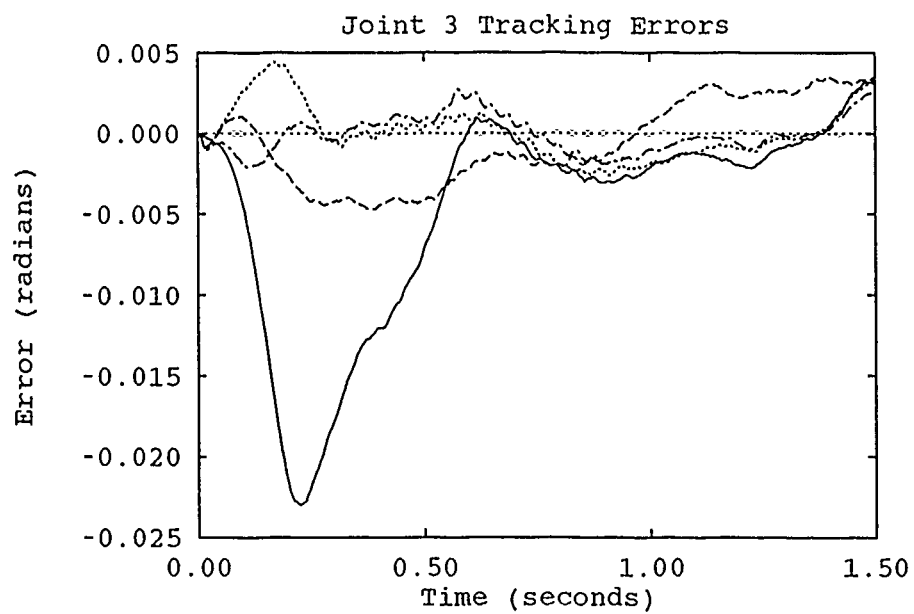


Figure C.18. 19-Parameter Uninitialized Adaptation Runs - Trajectory 1

—	First Run	Sixth Run
- - -	Third Run	- . - .	Ninth Run

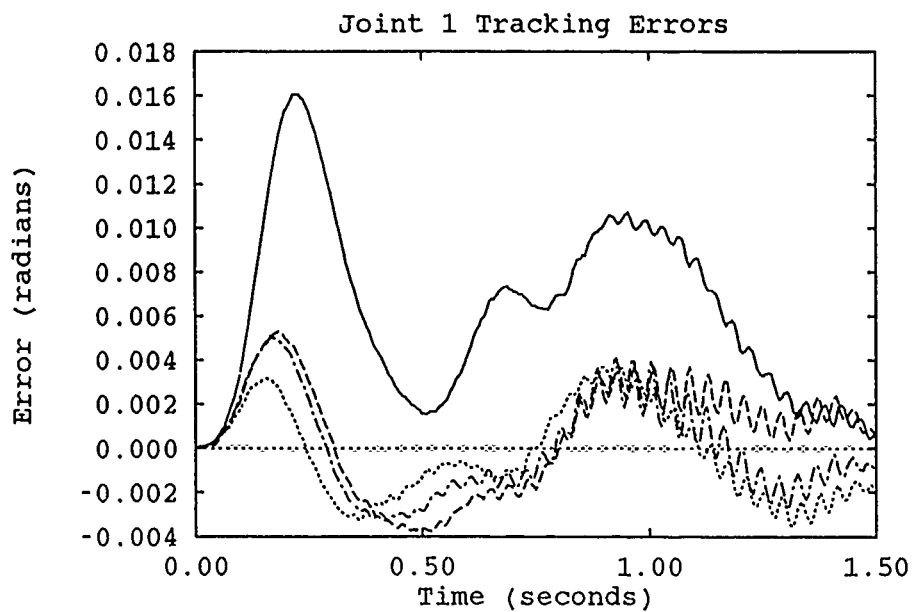


Figure C.19. 19 Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

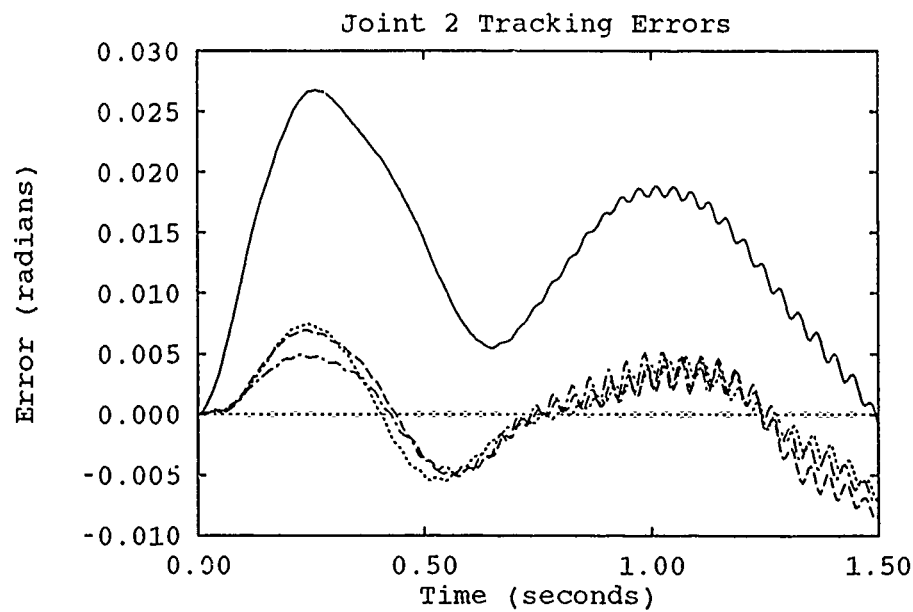


Figure C.20. 19-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

—	First Run	Sixth Run
---	Third Run	-.-.-	Ninth Run

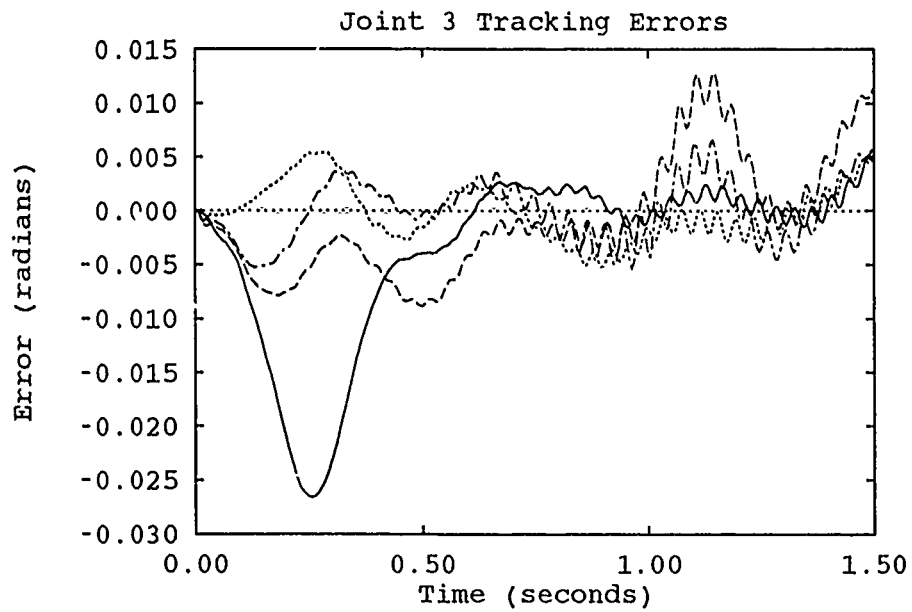


Figure C.21. 19-Parameter Uninitialized Adaptation Runs - Trajectory 1 w/ Payload

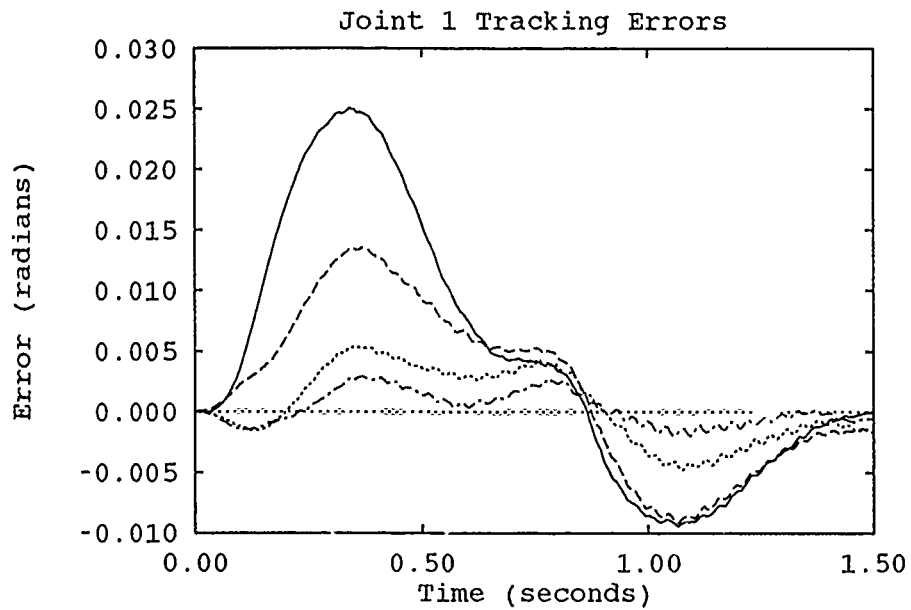


Figure C.22. 19-Parameter Uninitialized Adaptation Runs - Trajectory 6

—	First Run	Sixth Run
---	Third Run	-.-.-.-	Ninth Run

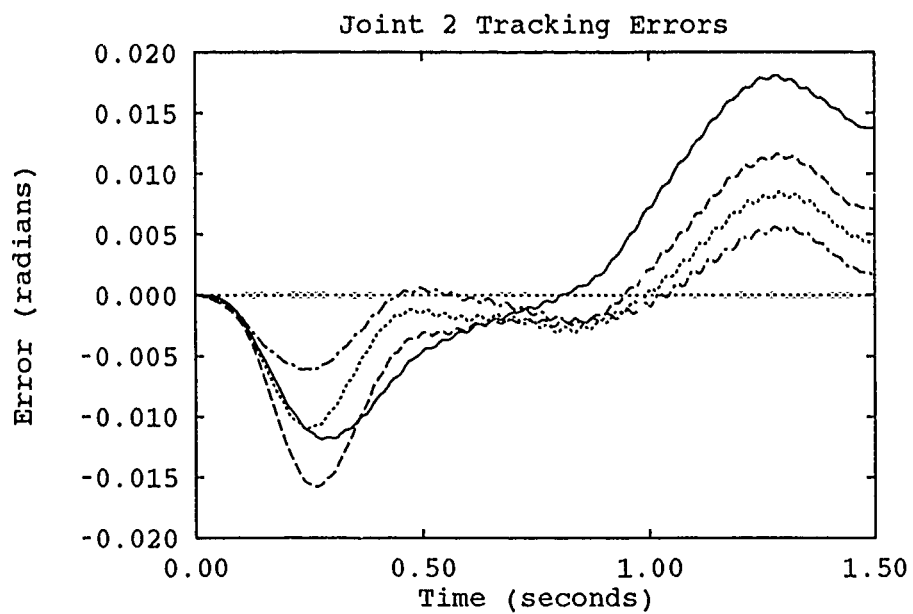


Figure C.23. 19-Parameter Uninitialized Adaptation Runs - Trajectory 6

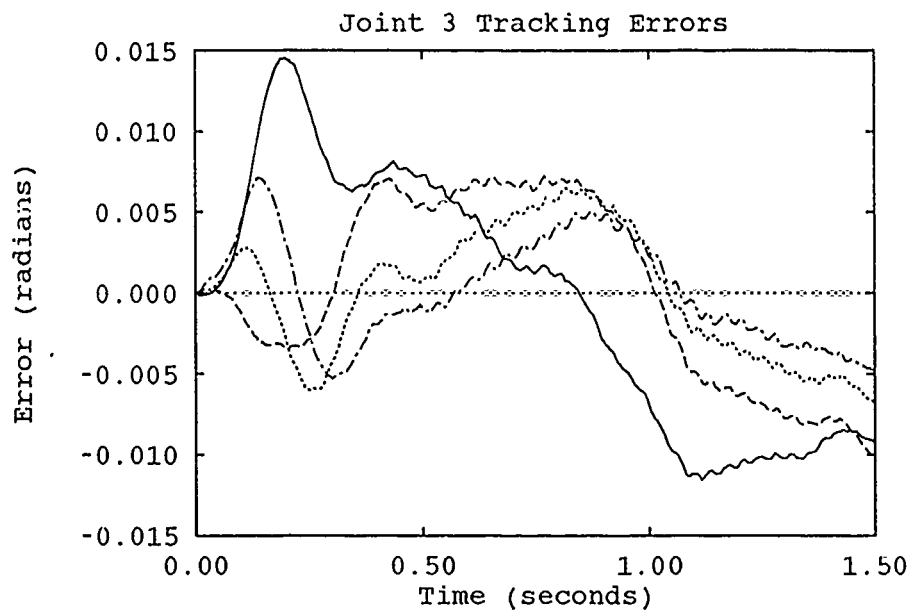


Figure C.24. 19-Parameter Uninitialized Adaptation Runs - Trajectory 6

—	First Run	Sixth Run
---	Third Run	-.-.-.	Ninth Run

Appendix D. *Comparisons of Single Uninitialized Runs*

This section contains a comparison of the first runs for 13-, 16-, and 19-parameter AMBC controllers for each test trajectory using uninitialized parameters.

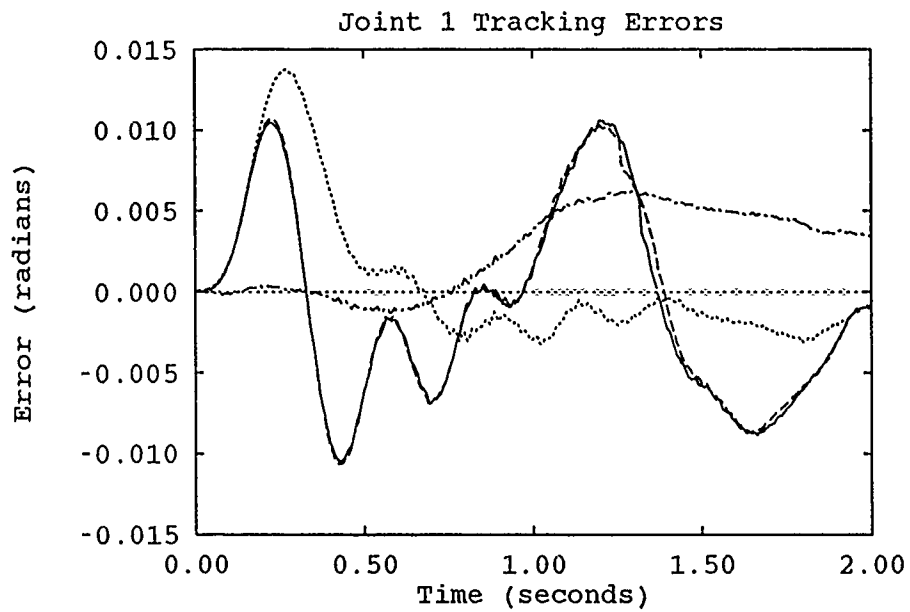


Figure D.1. Comparison of Single Uninitialized Runs - Trajectory 2

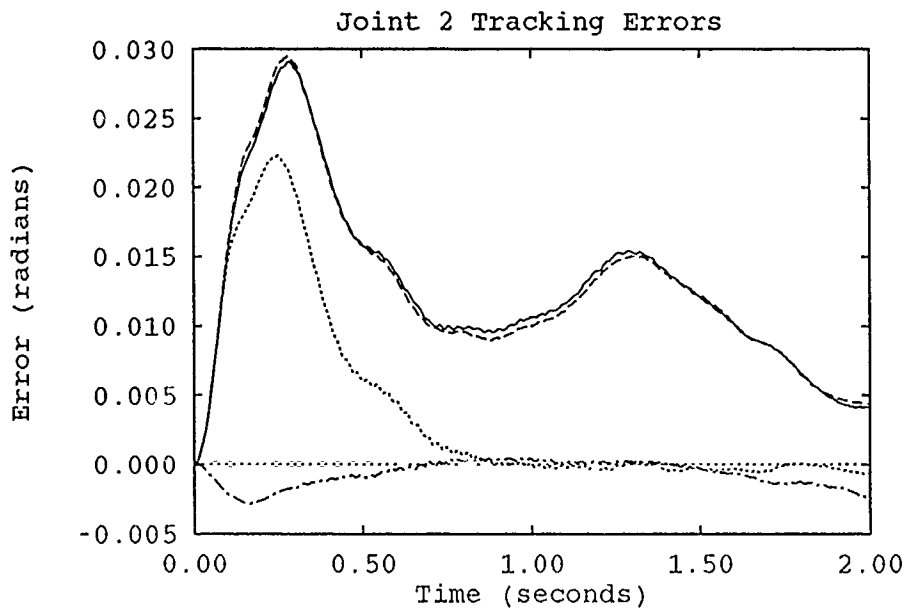


Figure D.2. Comparison of Single Uninitialized Runs - Trajectory 2

—	19 Parameters	13 Parameters
- - - -	16 Parameters	- . - . -	SMBC

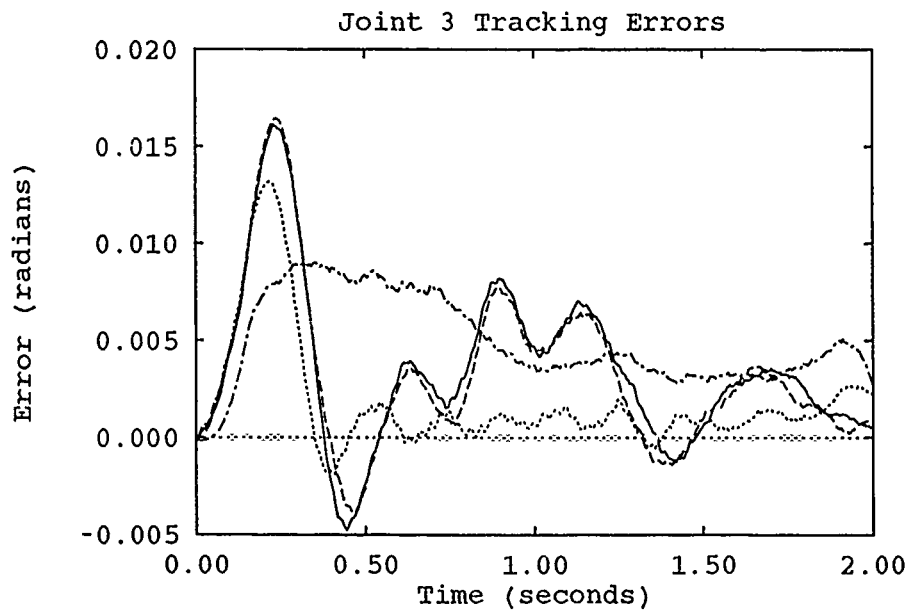


Figure D.3. Comparison of Single Uninitialized Runs - Trajectory 2

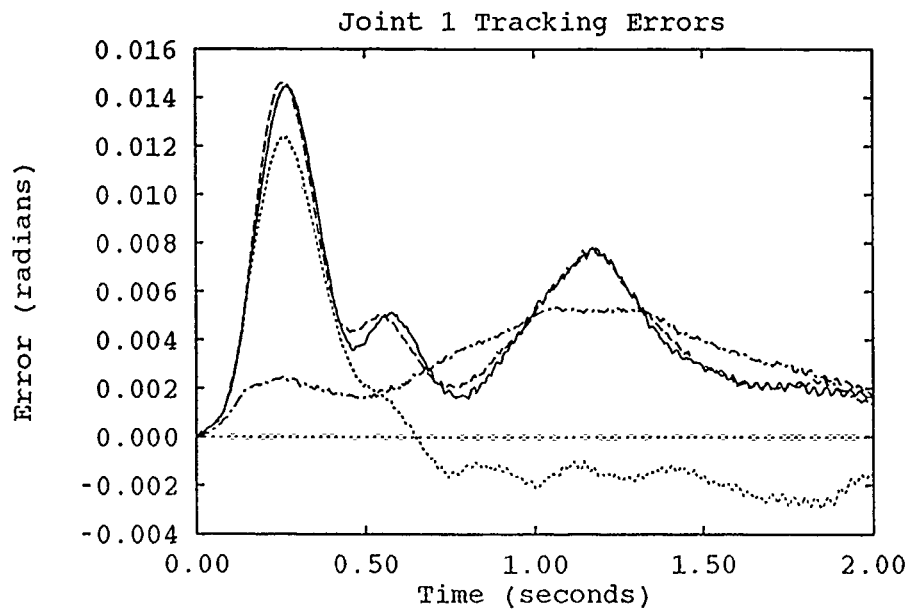


Figure D.4. Comparison of Single Uninitialized Runs - Trajectory 3

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - . -	SMBC

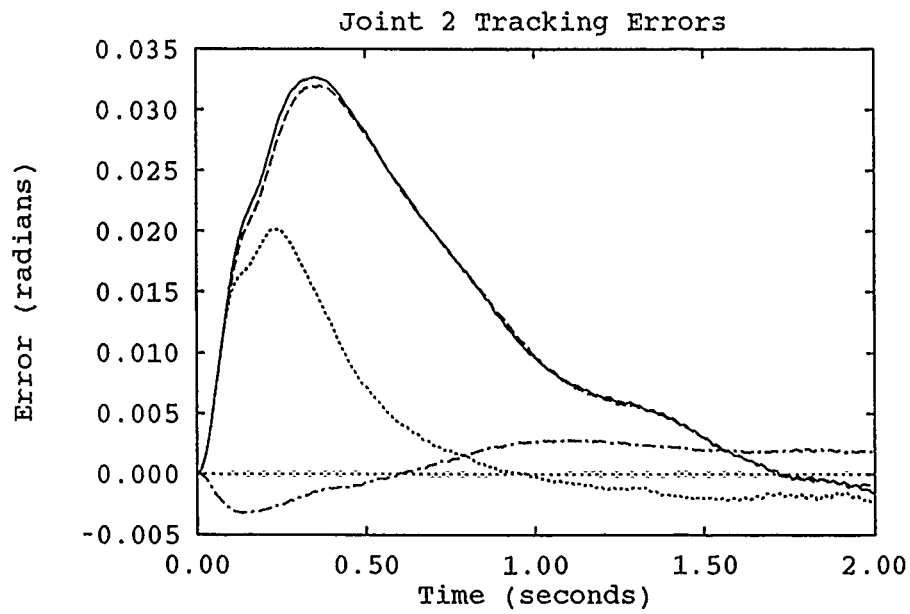


Figure D.5. Comparison of Single Uninitialized Runs - Trajectory 3

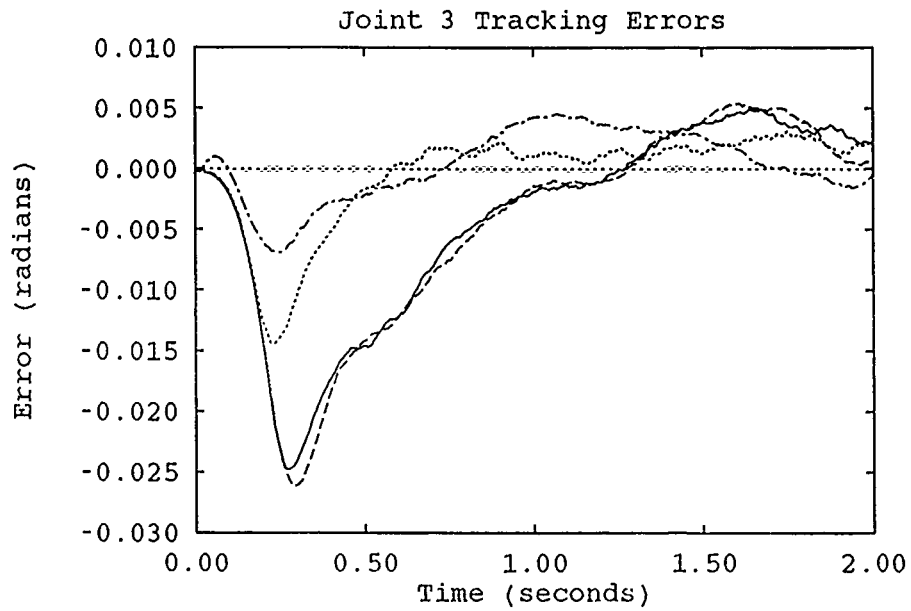


Figure D.6. Comparison of Single Uninitialized Runs - Trajectory 3

—	19 Parameters	13 Parameters
- - -	16 Parameters	- - - - -	SMBC

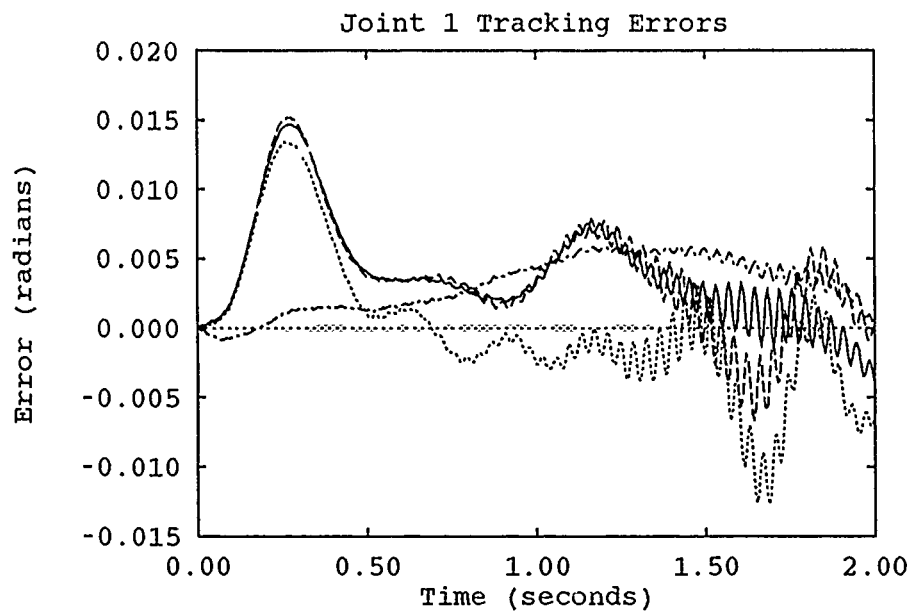


Figure D.7. Comparison of Single Uninitialized Runs - Trajectory 3 w/ Payload

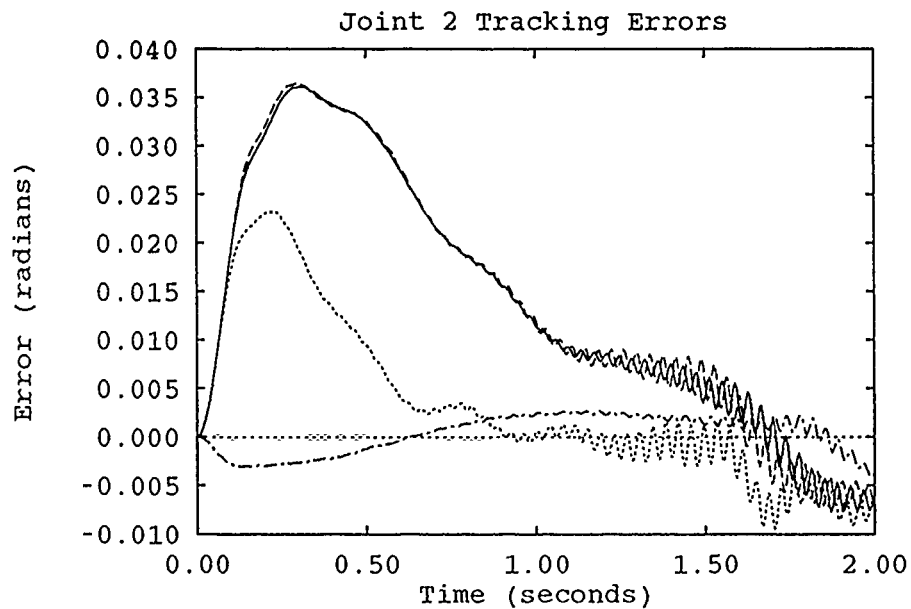


Figure D.8. Comparison of Single Uninitialized Runs - Trajectory 3 w/ Payload

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

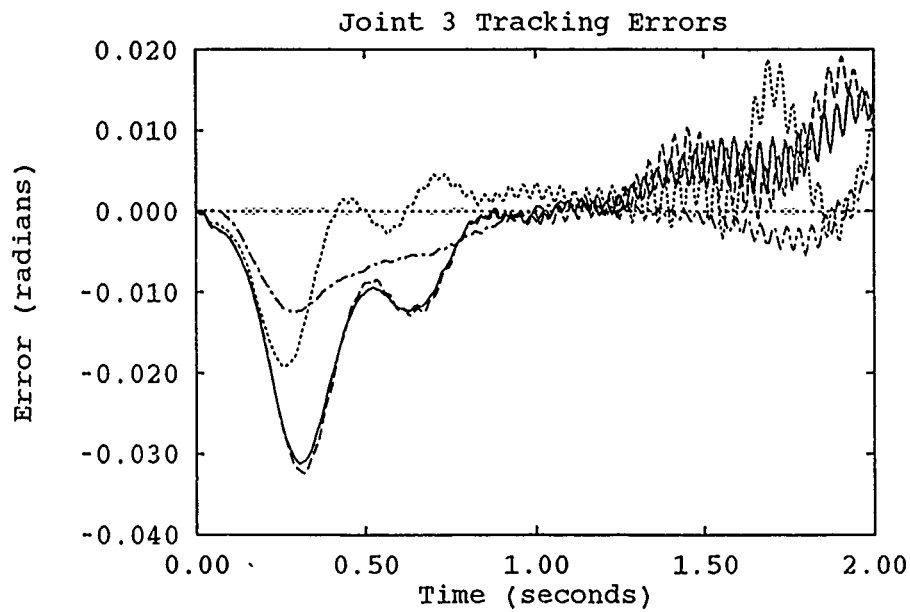


Figure D.9. Comparison of Single Uninitialized Runs - Trajectory 3 w/ Payload

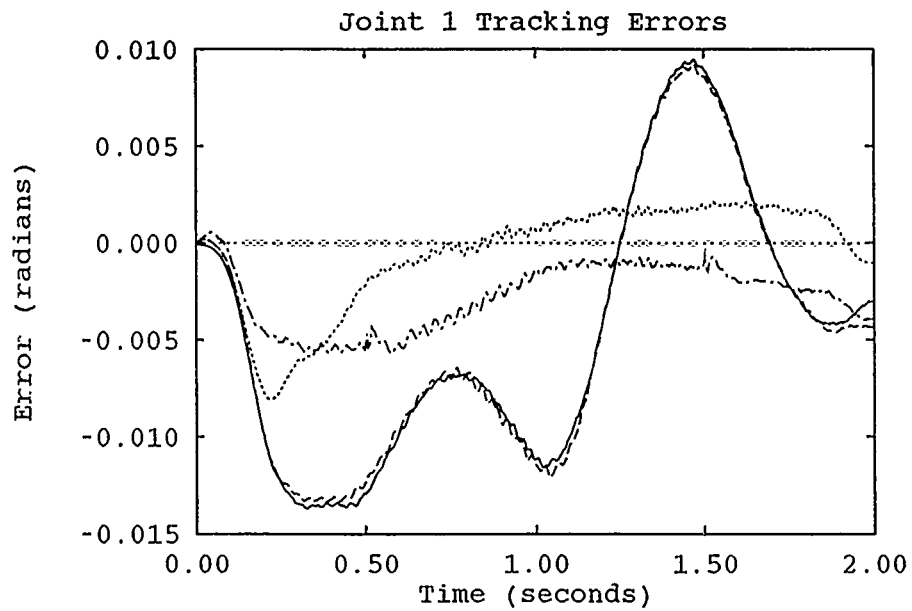


Figure D.10. Comparison of Single Uninitialized Runs - Trajectory 4

—	19 Parameters	13 Parameters
---	16 Parameters	-.-.-	SMBC

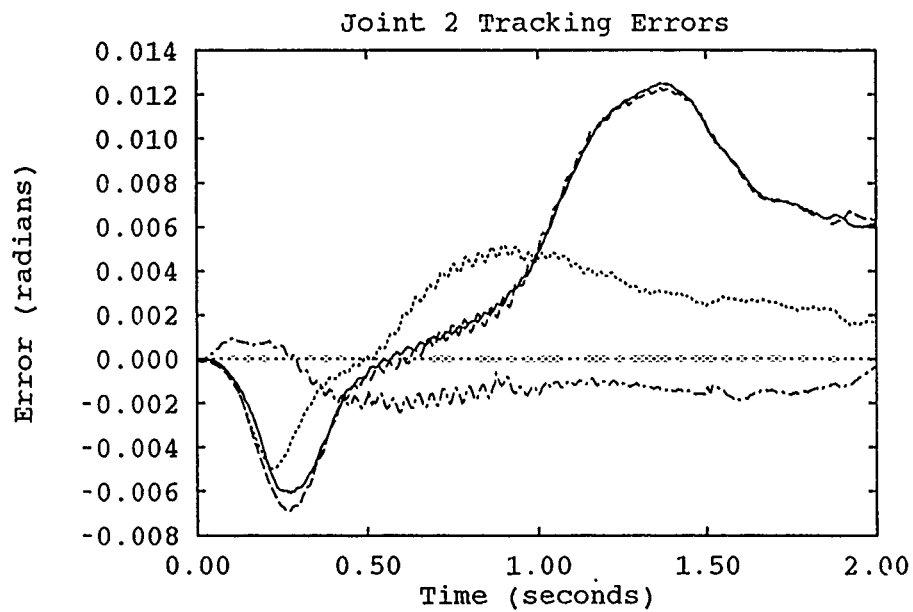


Figure D.11. Comparison of Single Uninitialized Runs - Trajectory 4

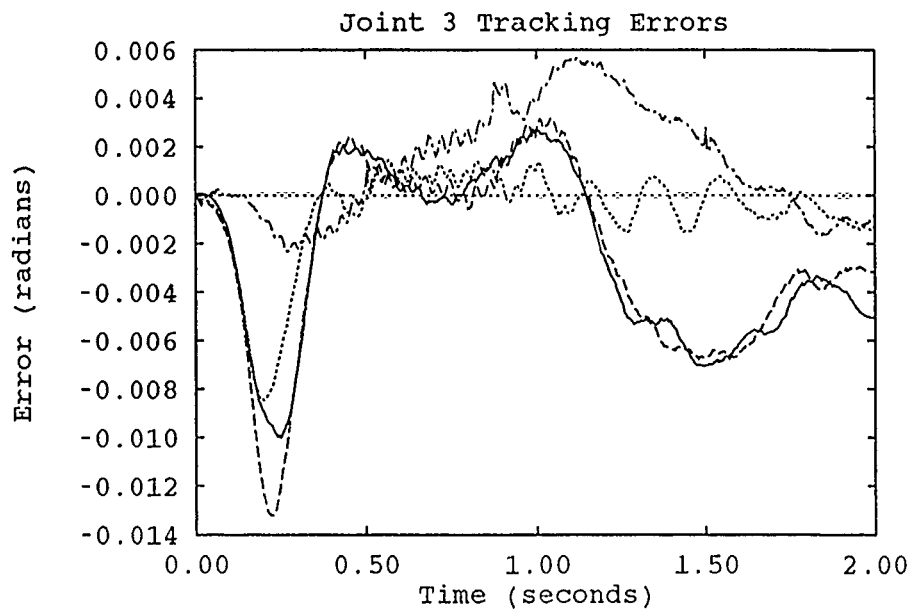


Figure D.12. Comparison of Single Uninitialized Runs - Trajectory 4

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - . -	SMBC

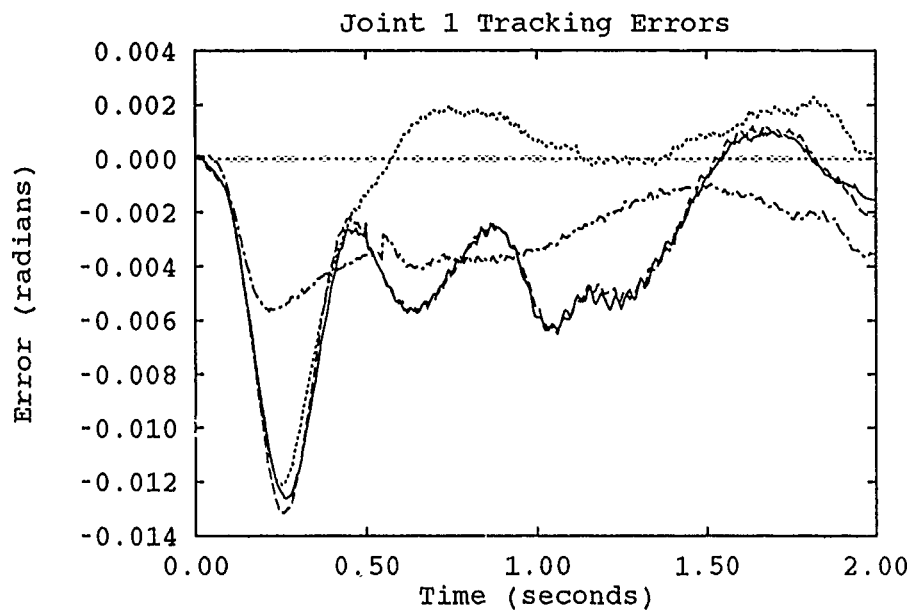


Figure D.13. Comparison of Single Uninitialized Runs - Trajectory 5

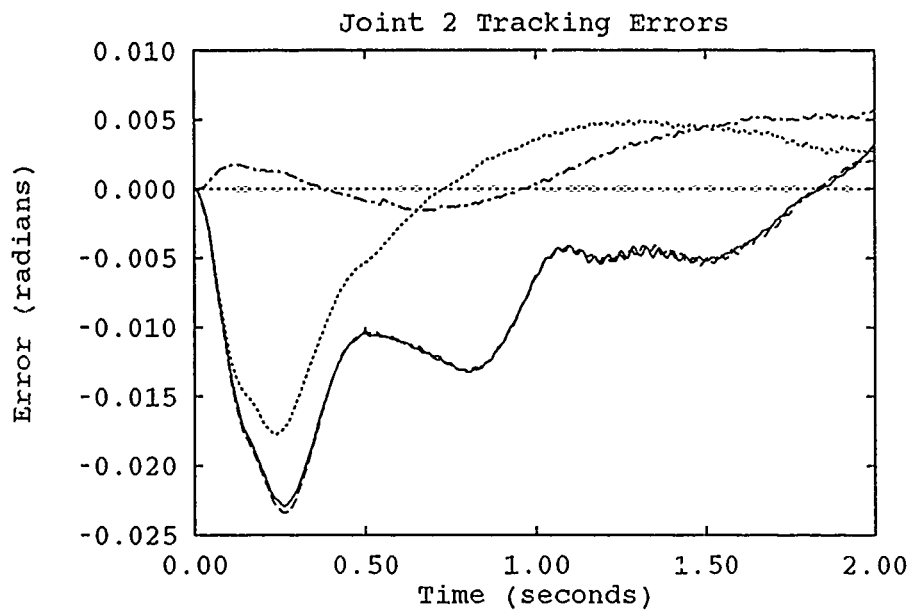


Figure D.14. Comparison of Single Uninitialized Runs - Trajectory 5

—	19 Parameters	13 Parameters
---	16 Parameters	-.-.-.-	SMBC

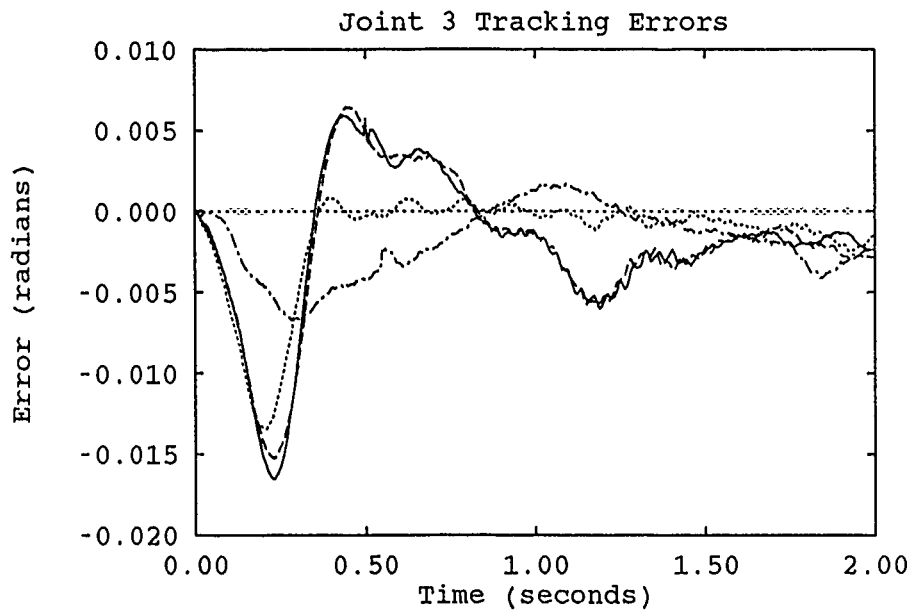


Figure D.15. Comparison of Single Uninitialized Runs - Trajectory 5

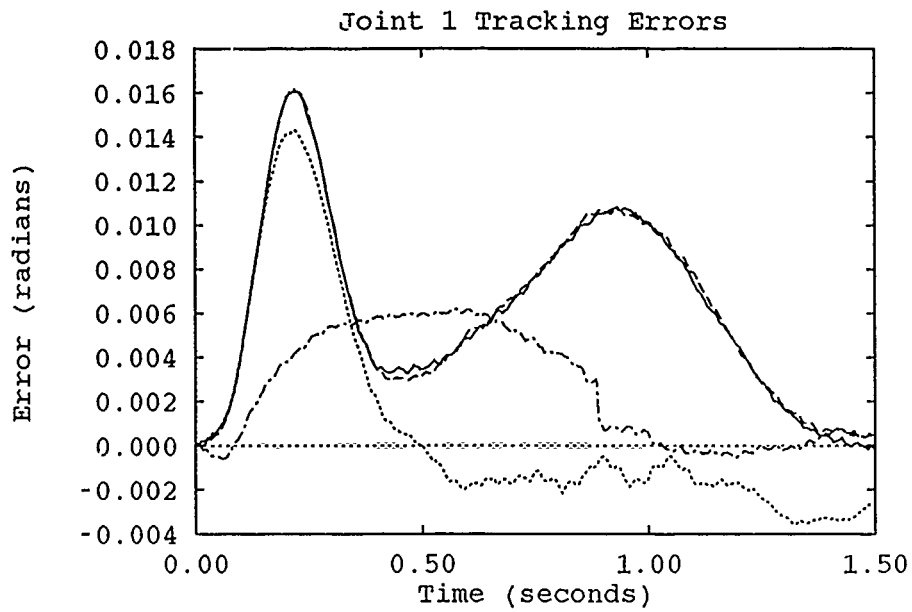


Figure D.16. Comparison of Single Uninitialized Runs - Trajectory 1

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

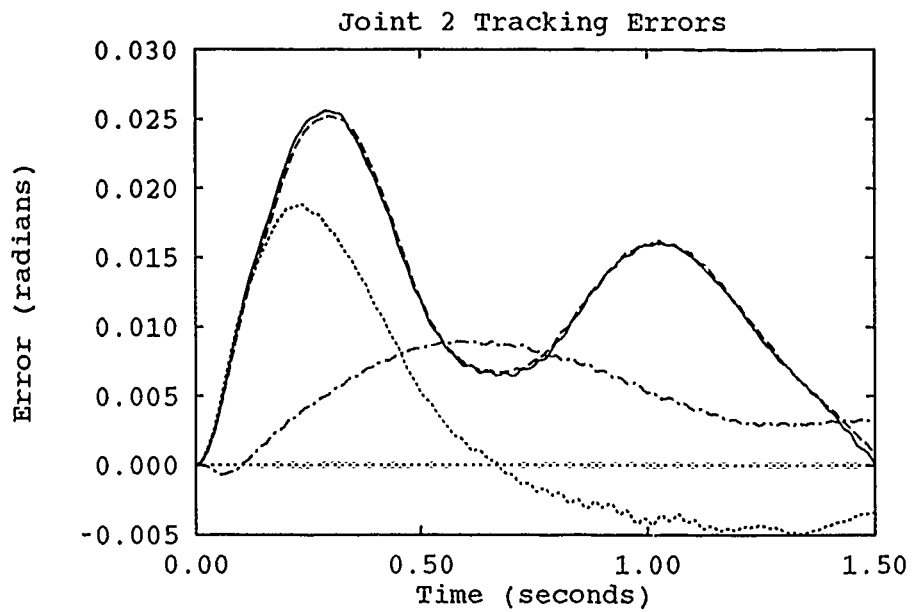


Figure D.17. Comparison of Single Uninitialized Runs - Trajectory 1

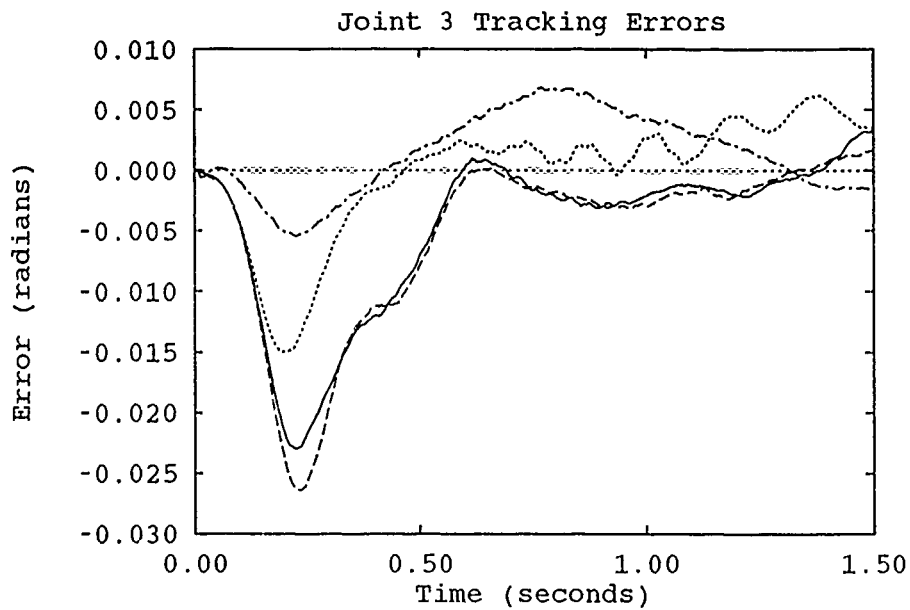


Figure D.18. Comparison of Single Uninitialized Runs - Trajectory 1

—	19 Parameters	13 Parameters
----	16 Parameters	- - - -	SMBC

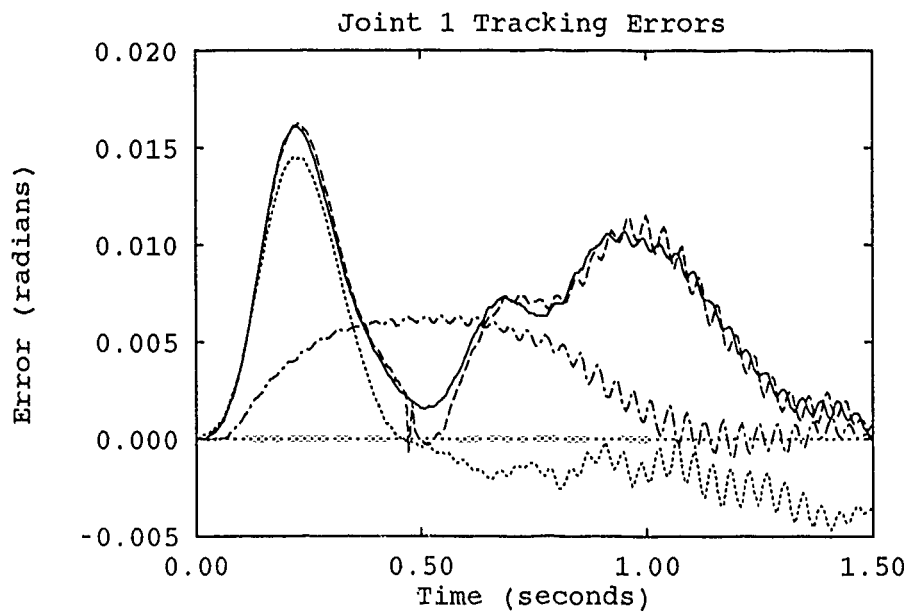


Figure D.19. Comparison of Single Uninitialized Runs - Trajectory 1 w/ Payload

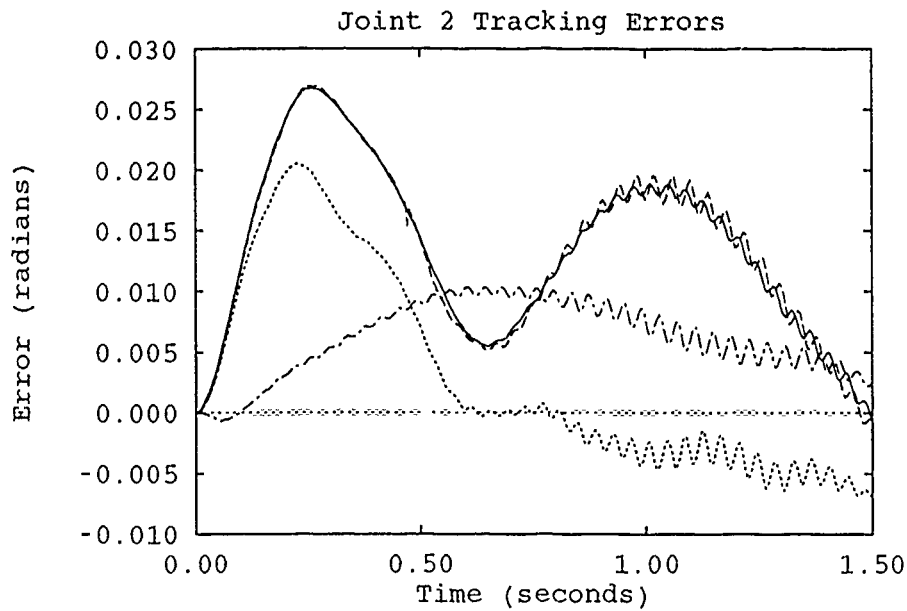


Figure D.20. Comparison of Single Uninitialized Runs - Trajectory 1 w/ Payload

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - . -	SMBC

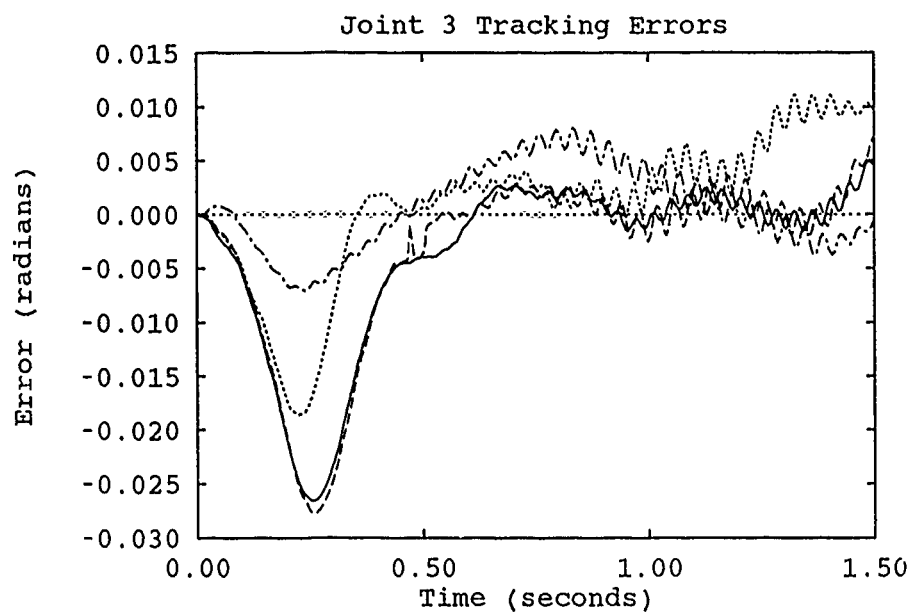


Figure D.21. Comparison of Single Uninitialized Runs - Trajectory 1 w/ Payload

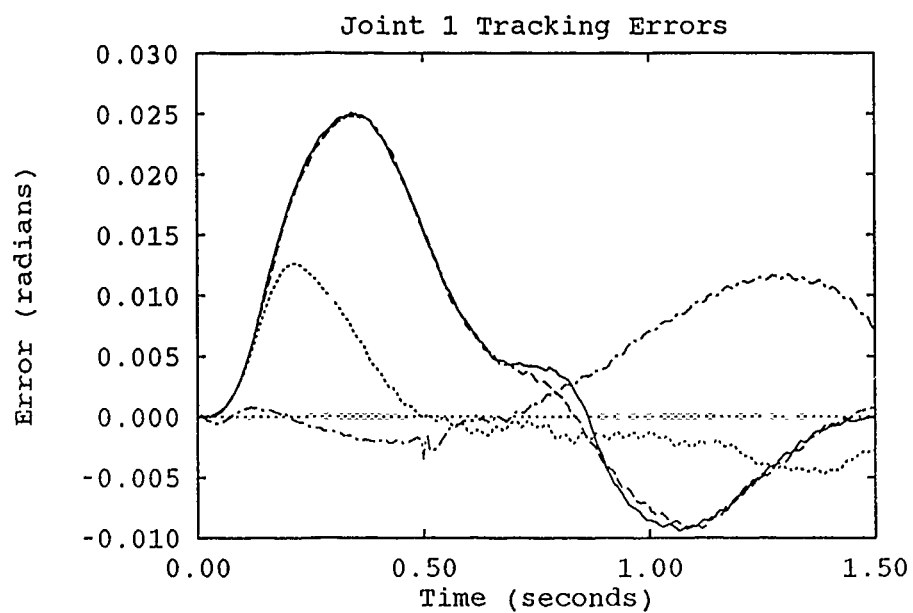


Figure D.22. Comparison of Single Uninitialized Runs - Trajectory 6

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

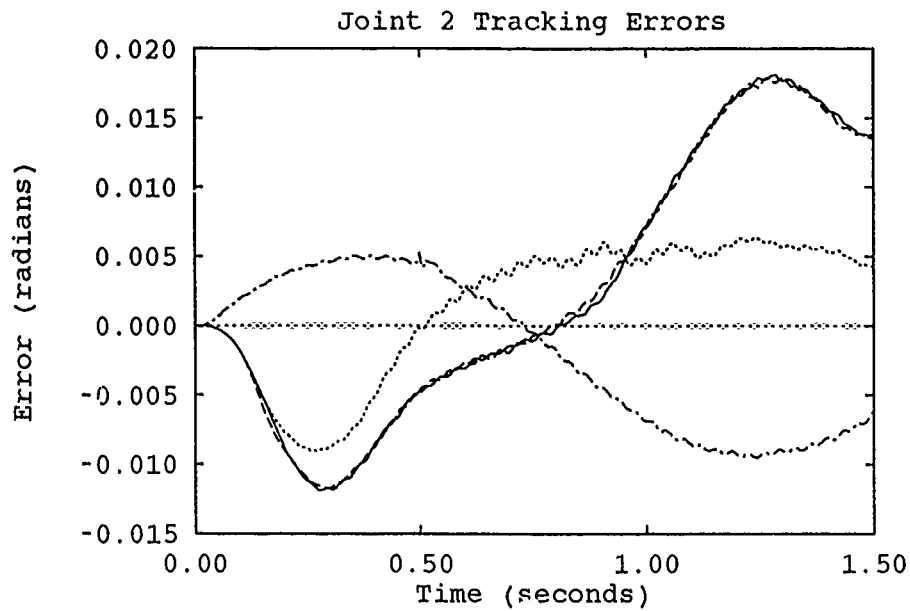


Figure D.23. Comparison of Single Uninitialized Runs - Trajectory 6

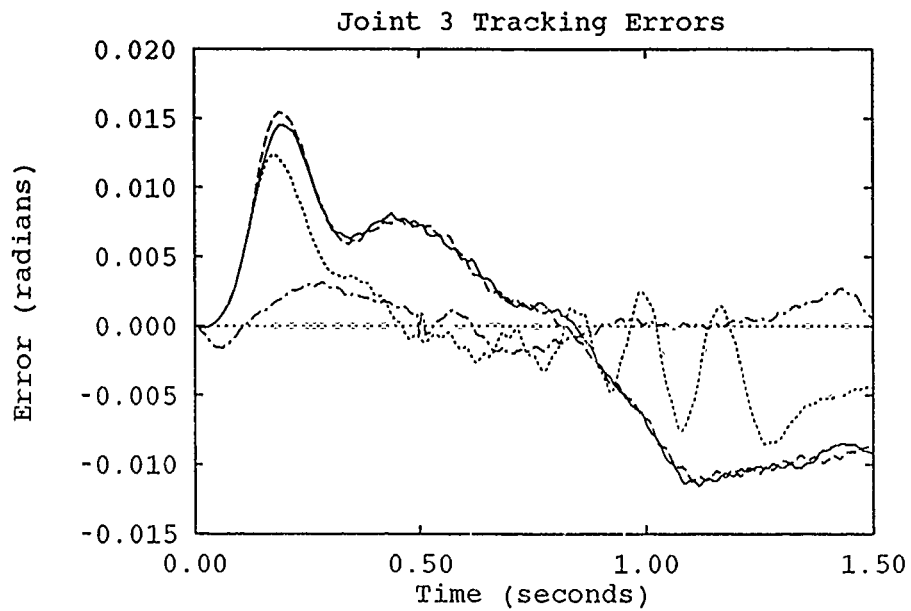


Figure D.24. Comparison of Single Uninitialized Runs - Trajectory 6

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

Appendix E. *Comparisons of Uninitialized Runs After Learning*

This section contains a comparison of the 13-, 16-, and 19-parameter adaptation runs for each test trajectory using uninitialized parameters. Each trajectory was run nine times to allow the controller to adapt.

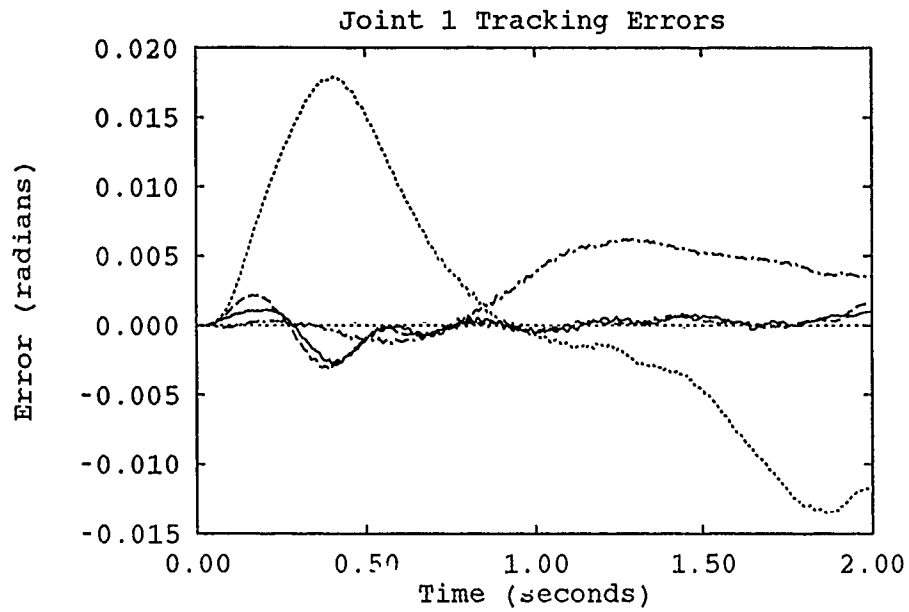


Figure E.1. Comparison of Uninitialized Runs After Learning - Trajectory 2

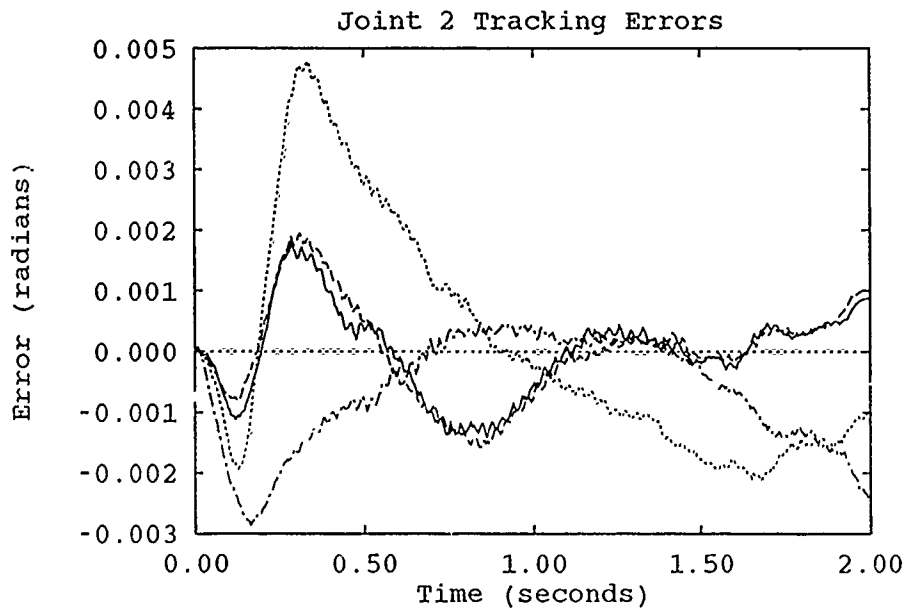


Figure E.2. Comparison of Uninitialized Runs After Learning Trajectory 2

—	19 Parameters	13 Parameters
----	16 Parameters	- - -	SMBC

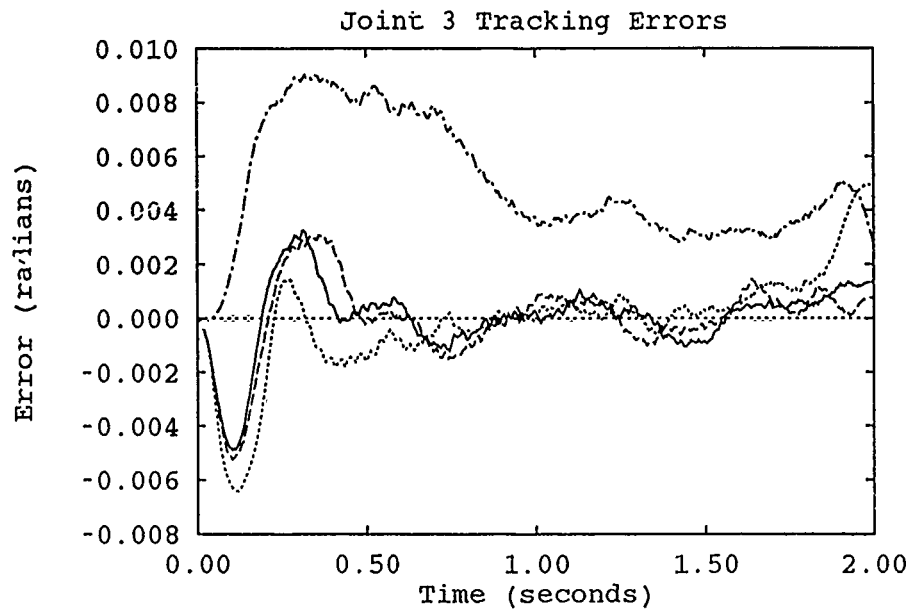


Figure E.3. Comparison of Uninitialized Runs After Learning - Trajectory 2

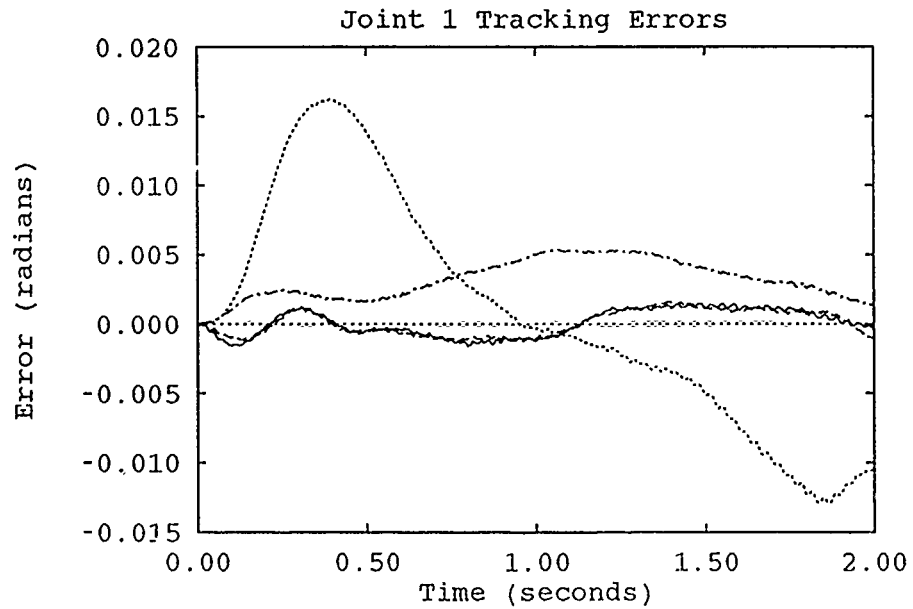


Figure E.4. Comparison of Uninitialized Runs After Learning - Trajectory 3

—	19 Parameters	·····	13 Parameters
- - -	16 Parameters	- · - · -	SMBC

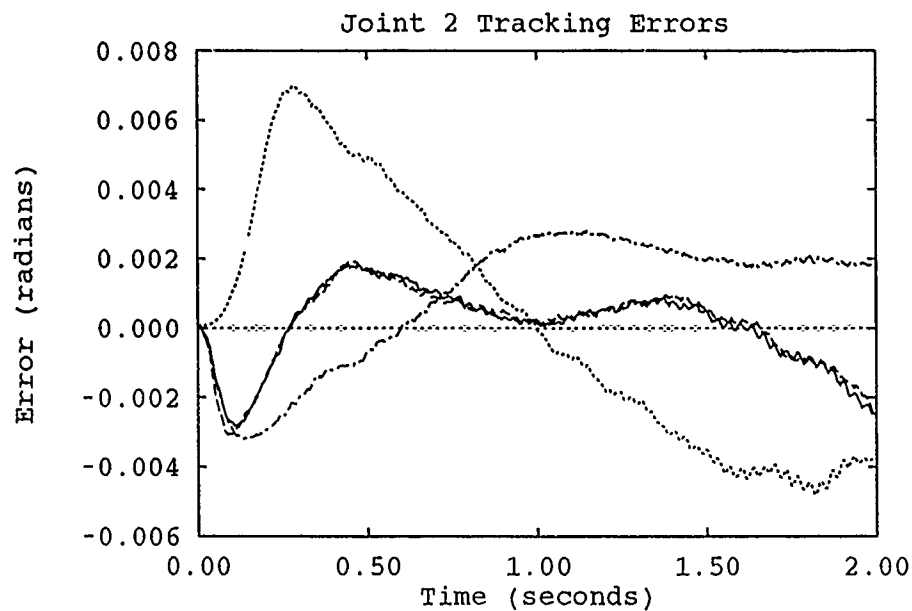


Figure E.5. Comparison of Uninitialized Runs After Learning - Trajectory 3

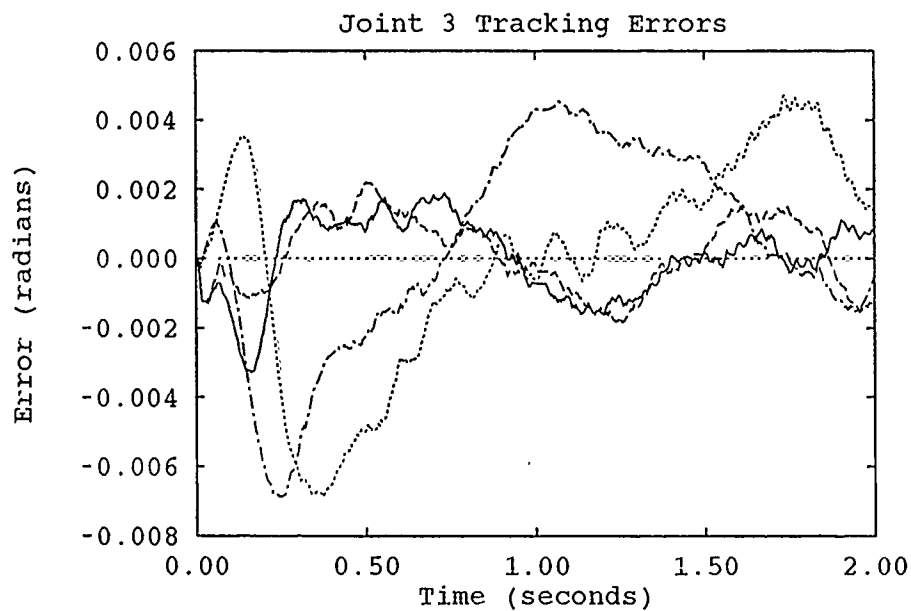


Figure E.6. Comparison of Uninitialized Runs After Learning - Trajectory 3

—	19 Parameters	13 Parameters
---	16 Parameters	-.-.-	SMBC

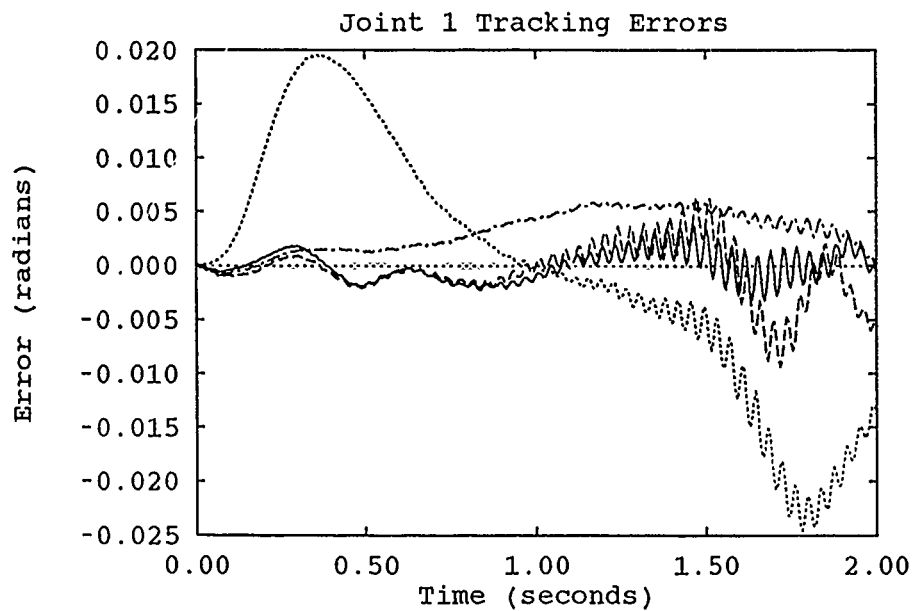


Figure E.7. Comparison of Uninitialized Runs After Learning - Trajectory 3 w/ Payload

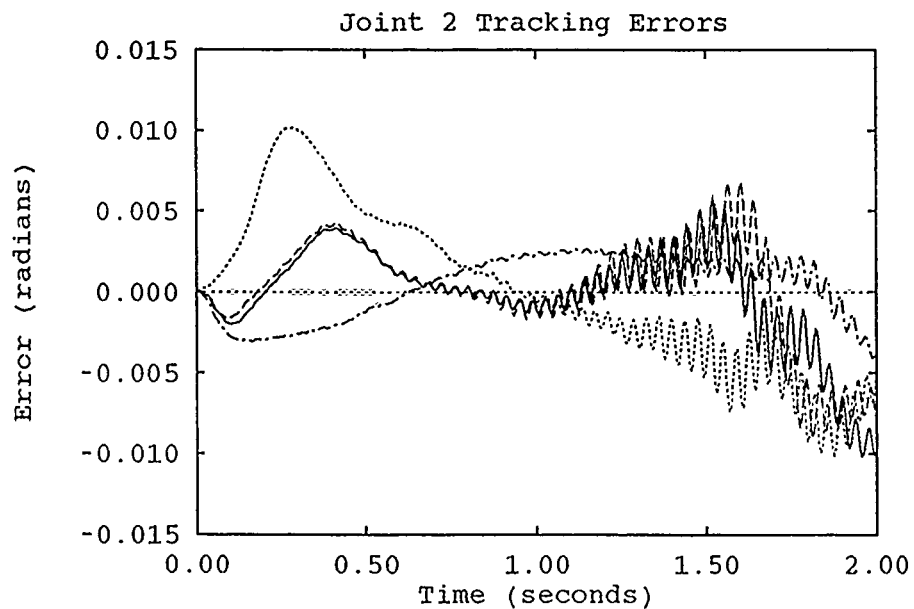


Figure E.8. Comparison of Uninitialized Runs After Learning - Trajectory 3 w/ Payload

—	19 Parameters	13 Parameters
---	16 Parameters	-.-.-	SMBC

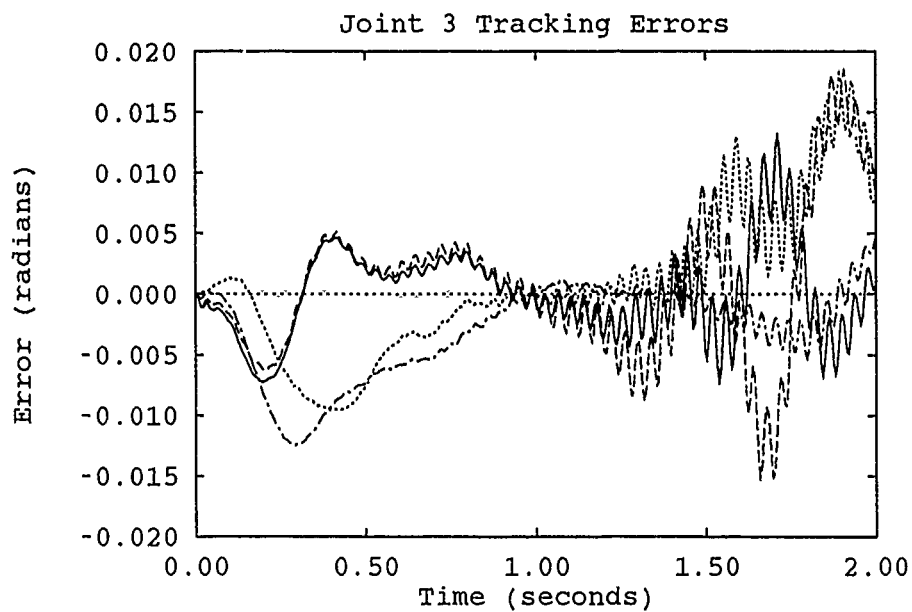


Figure E.9. Comparison of Uninitialized Runs After Learning - Trajectory 3 w/ Payload

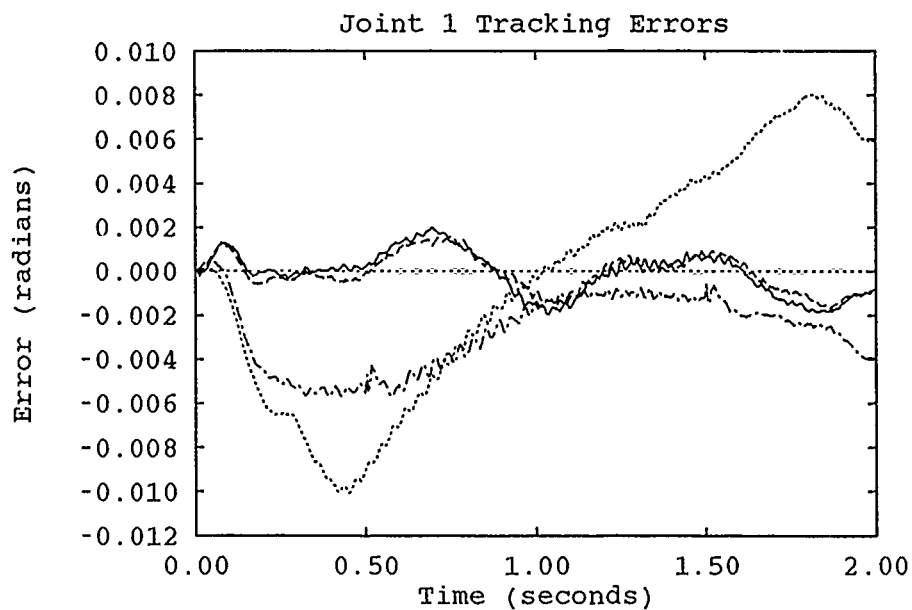


Figure E.10. Comparison of Uninitialized Runs After Learning - Trajectory 4

—	19 Parameters	13 Parameters
---	16 Parameters	-.-.-	SMBC

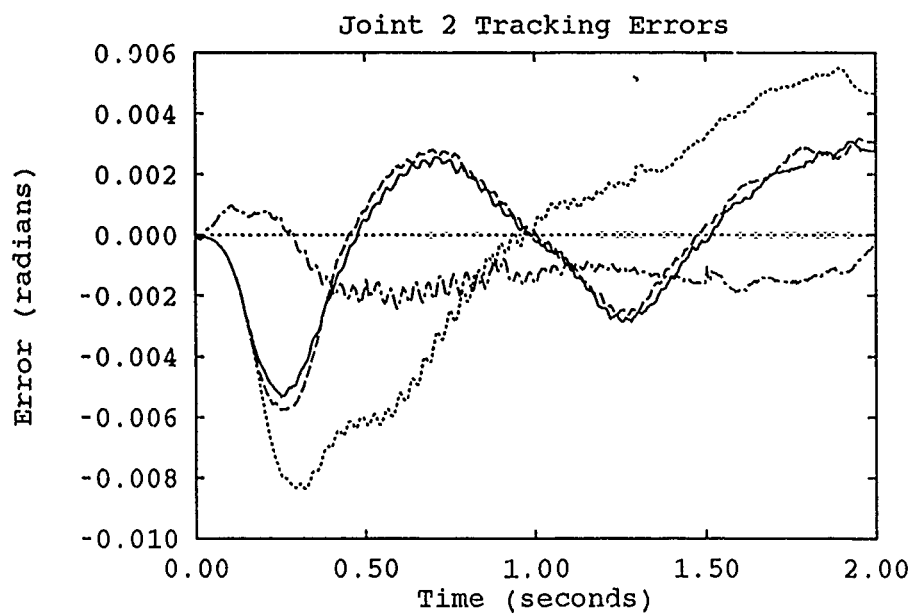


Figure E.11. Comparison of Uninitialized Runs After Learning - Trajectory 4

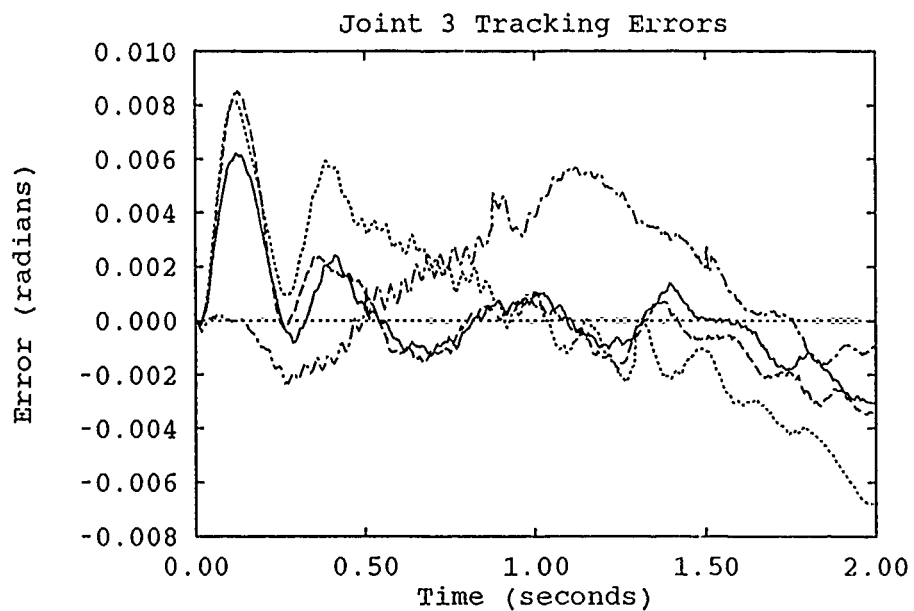


Figure E.12. Comparison of Uninitialized Runs After Learning - Trajectory 4

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

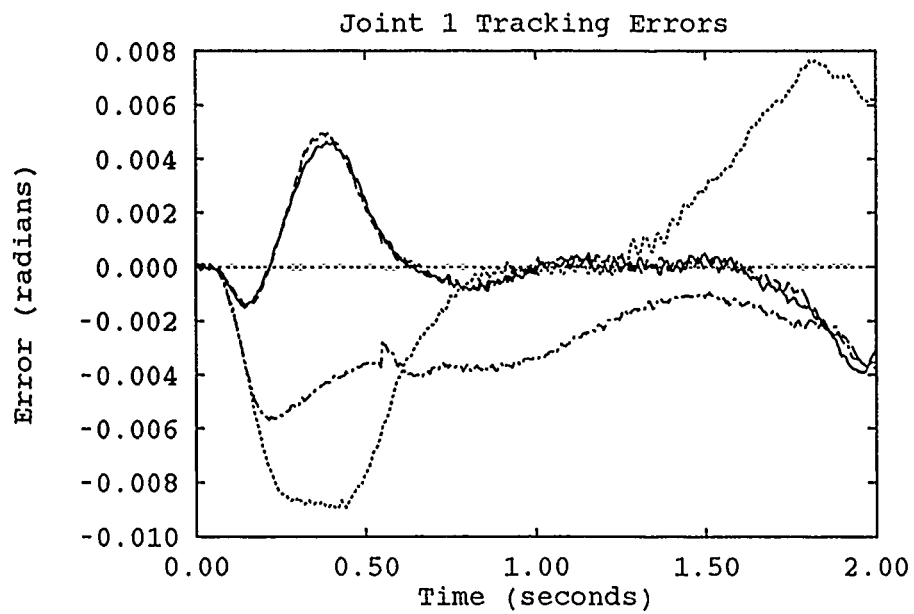


Figure E.13. Comparison of Uninitialized Runs After Learning - Trajectory 5

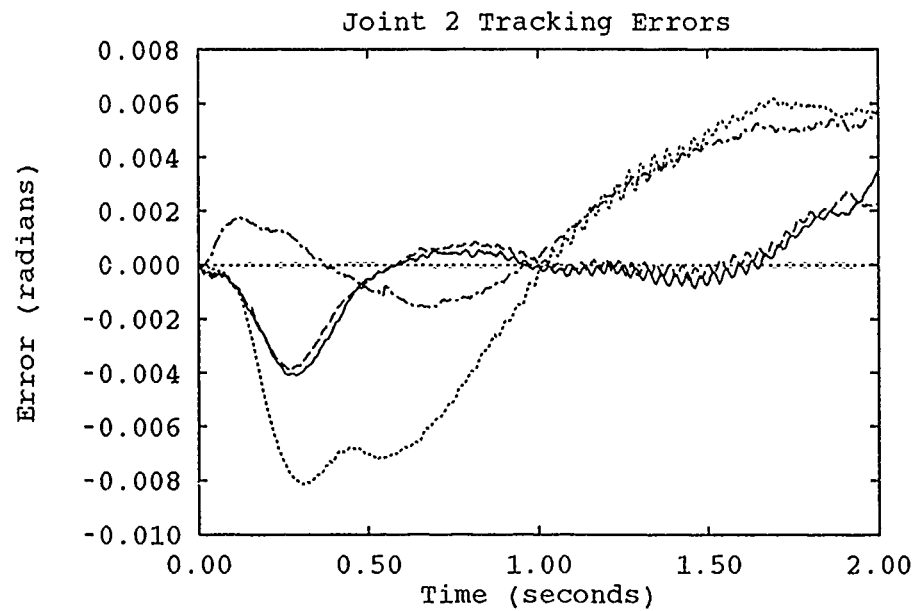


Figure E.14. Comparison of Uninitialized Runs After Learning - Trajectory 5

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - . -	SMBC

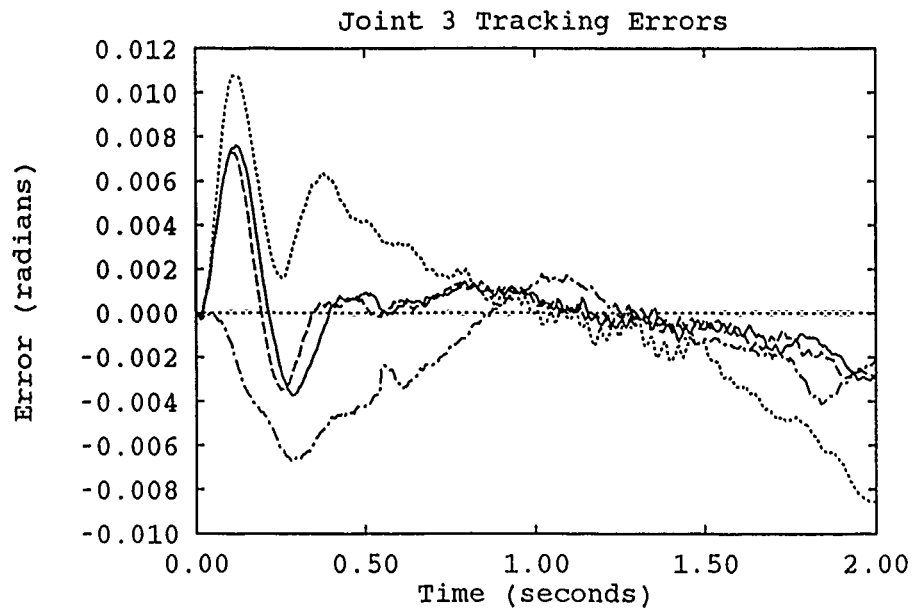


Figure E.15. Comparison of Uninitialized Runs After Learning - Trajectory 5

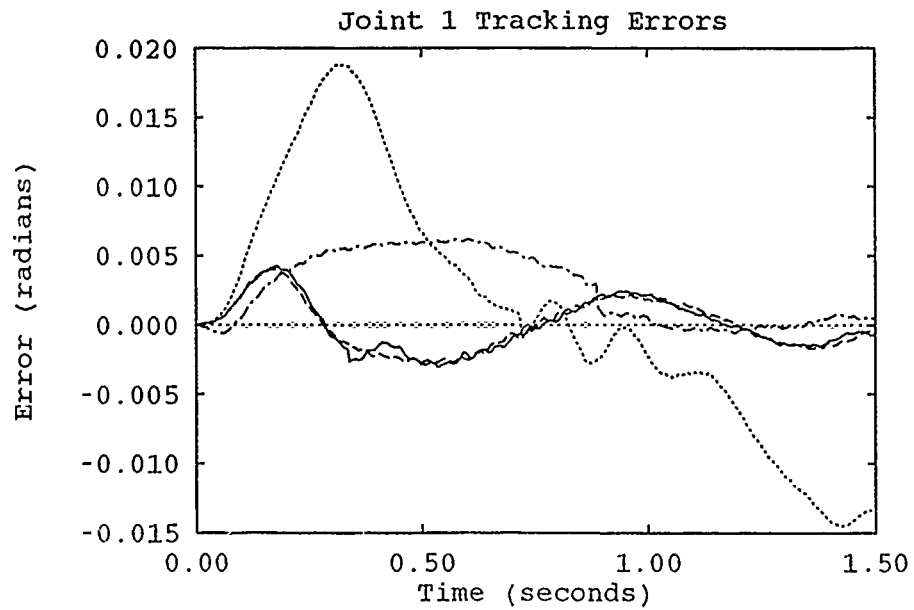


Figure E.16. Comparison of Uninitialized Runs After Learning - Trajectory 1

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

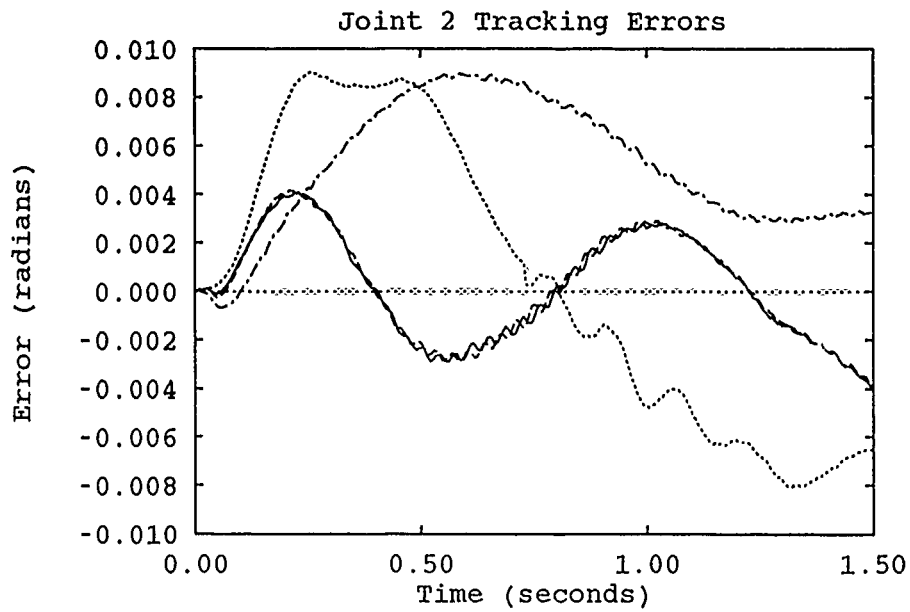


Figure E.17. Comparison of Uninitialized Runs After Learning - Trajectory 1

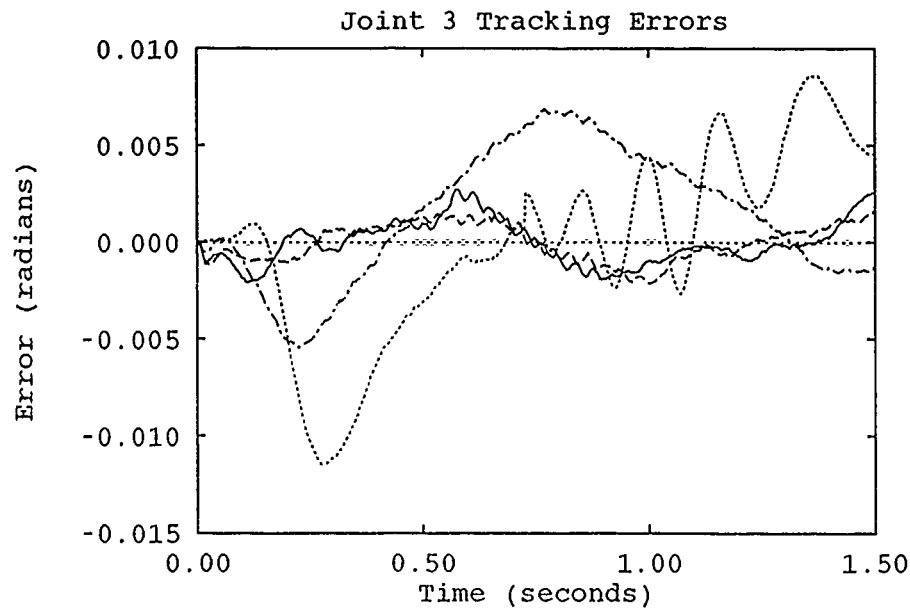


Figure E.18. Comparison of Uninitialized Runs After Learning - Trajectory 1

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

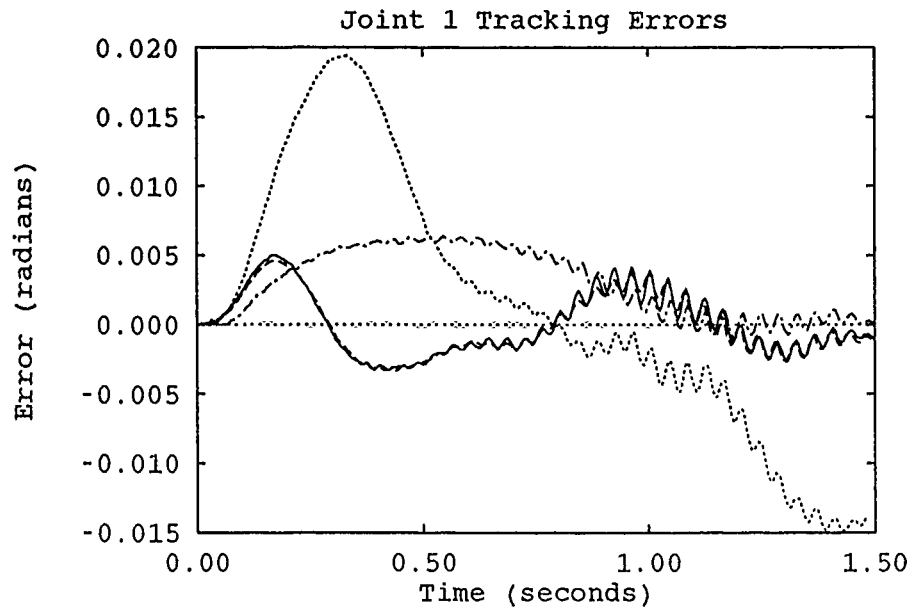


Figure E.19. Comparison of Uninitialized Runs After Learning - Trajectory 1 w/ Payload

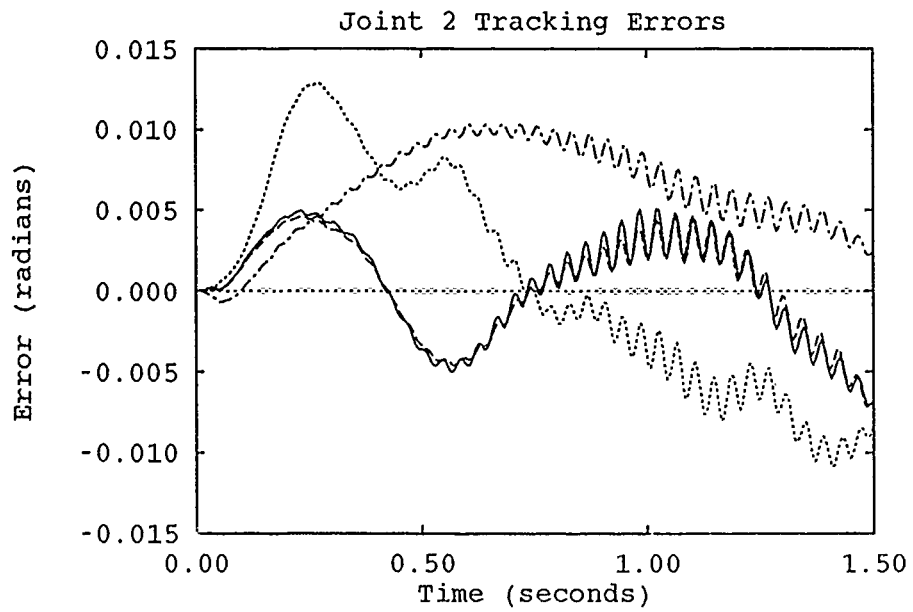


Figure E.20. Comparison of Uninitialized Runs After Learning - Trajectory 1 w/ Payload

—	19 Parameters	13 Parameters
---	16 Parameters	-.-.-	SMBC

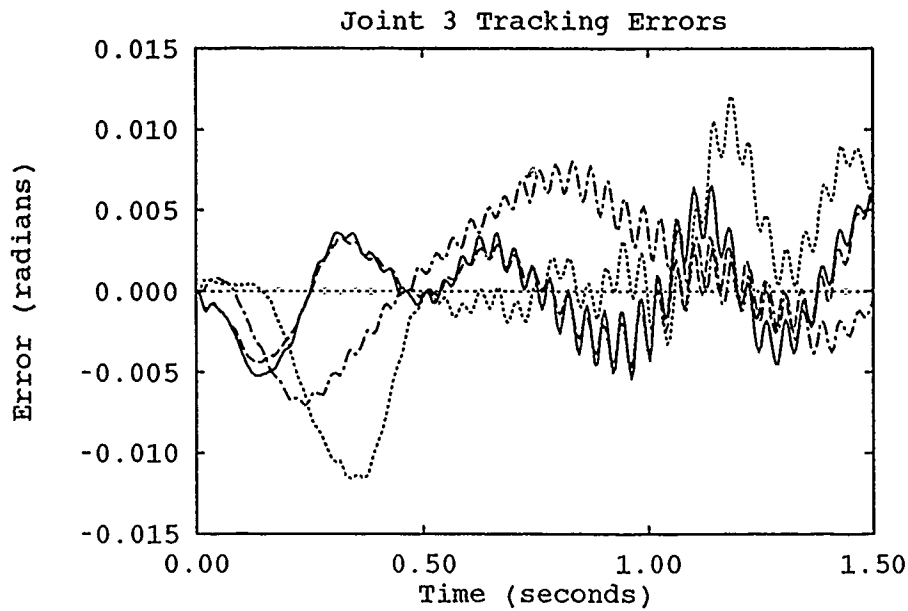


Figure E.21. Comparison of Uninitialized Runs After Learning - Trajectory 1 w/ Payload

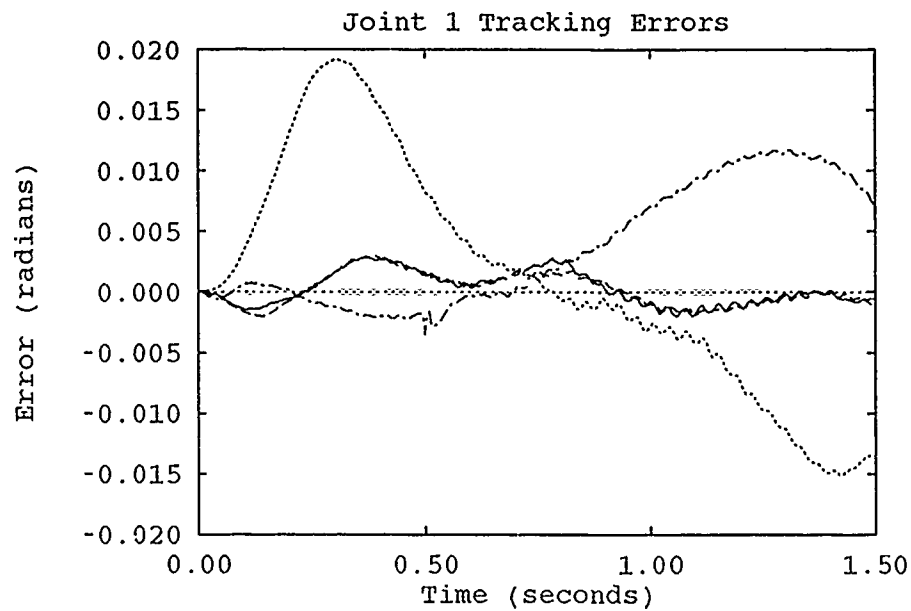


Figure E.22. Comparison of Uninitialized Runs After Learning - Trajectory 6

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

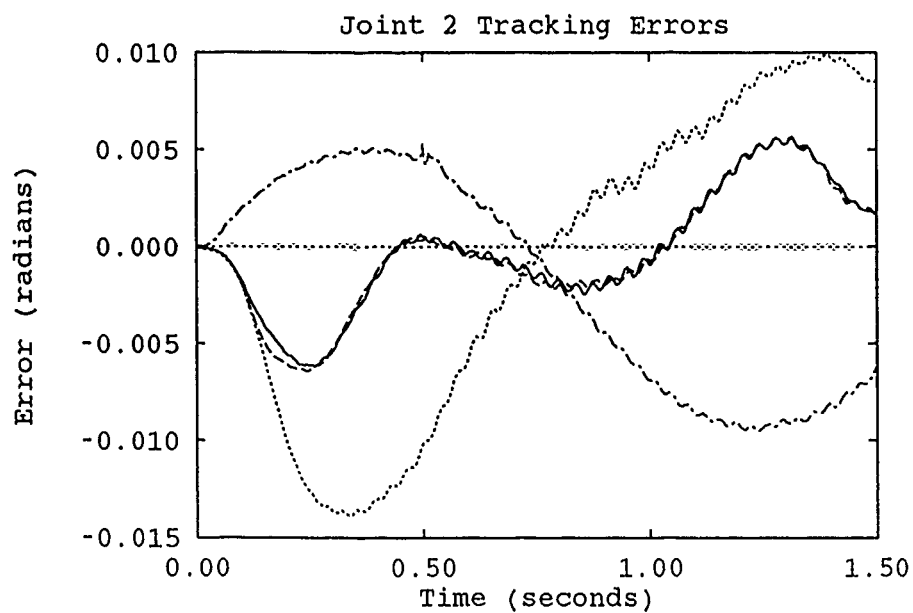


Figure E.23. Comparison of Uninitialized Runs After Learning - Trajectory 6

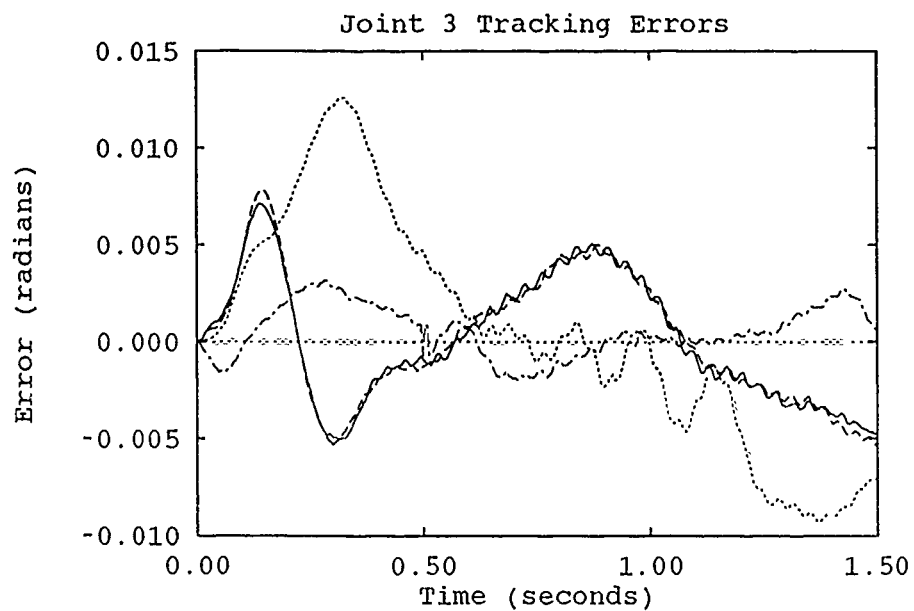


Figure E.24. Comparison of Uninitialized Runs After Learning - Trajectory 6

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

Appendix F. *13-Parameter Initialized Learning Runs*

This section contain the results of 13-parameter adaptation runs for each test trajectory using initialized parameters. Each trajectory was run five times to allow the controller to adapt.

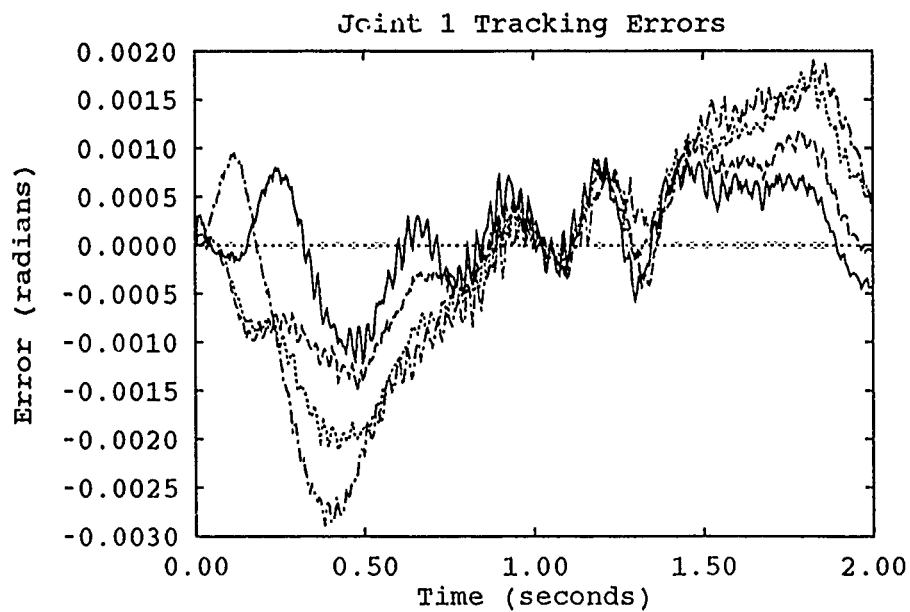


Figure F.1. 13-Parameter Initialized Adaptation Runs - Trajectory 2

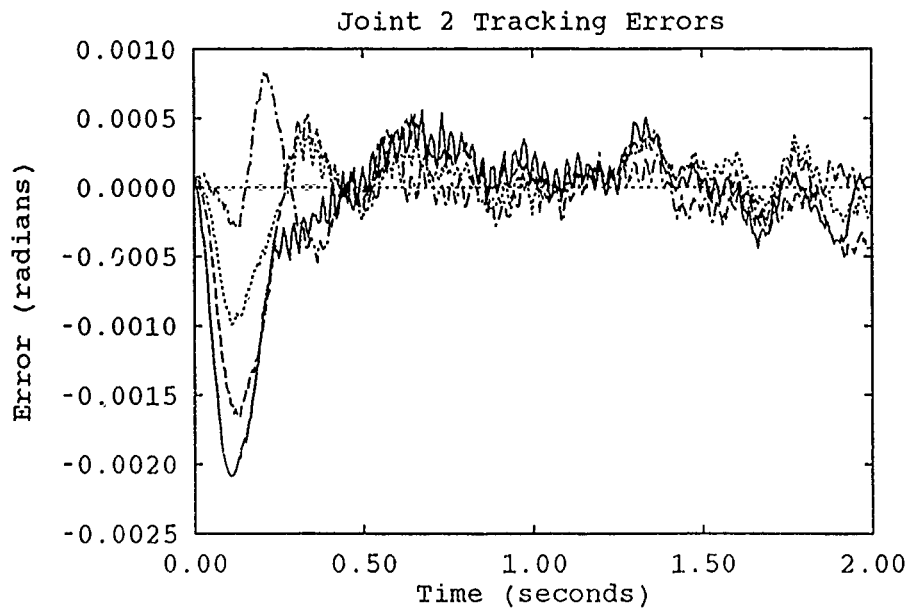


Figure F.2. 13-Parameter Initialized Adaptation Runs - Trajectory 2

—	First Run	Fourth Run
- - -	Second Run	- - - - -	Fifth Run

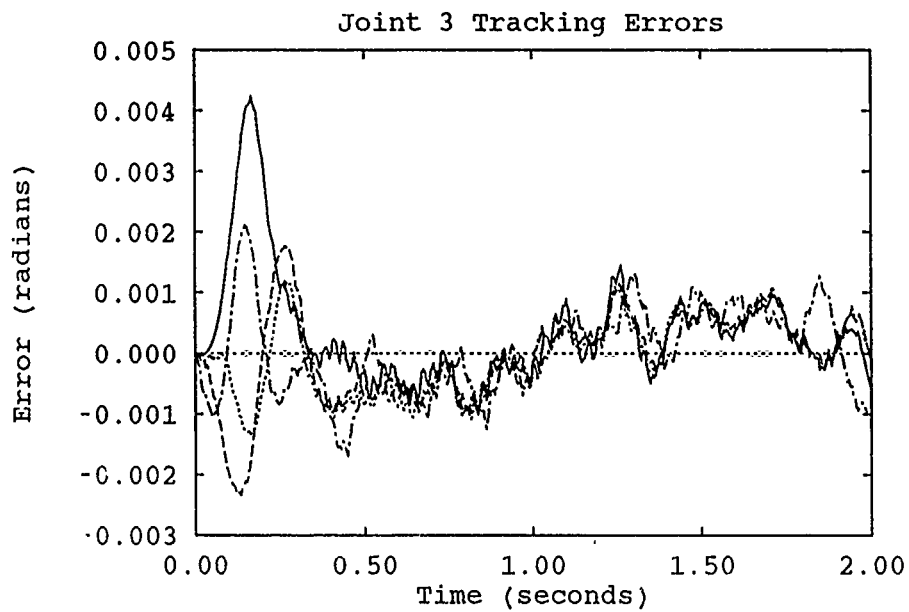


Figure F.3. 13-Parameter Initialized Adaptation Runs - Trajectory 2

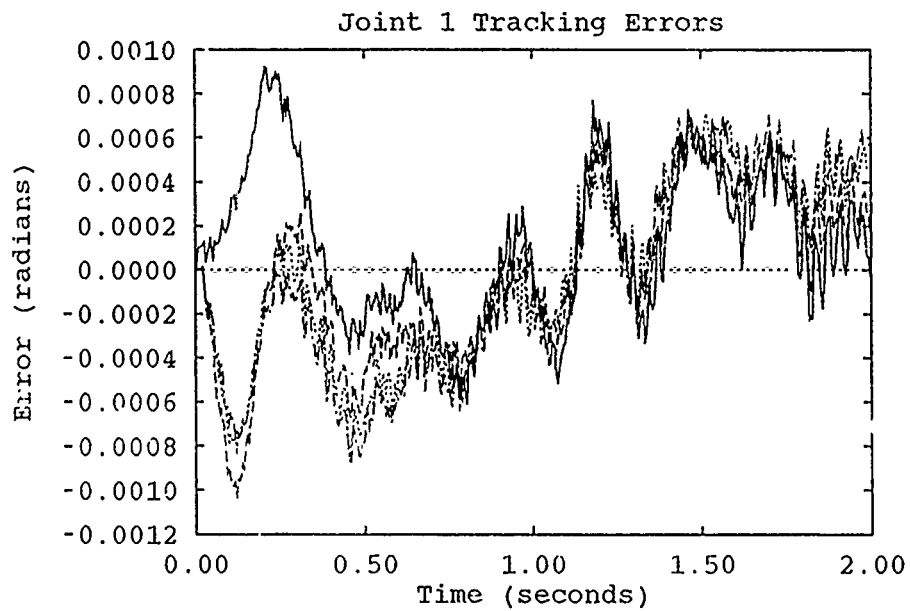


Figure F.4. 13-Parameter Initialized Adaptation Runs - Trajectory 3

—	First Run	Fourth Run
----	Second Run	- - - -	Fifth Run

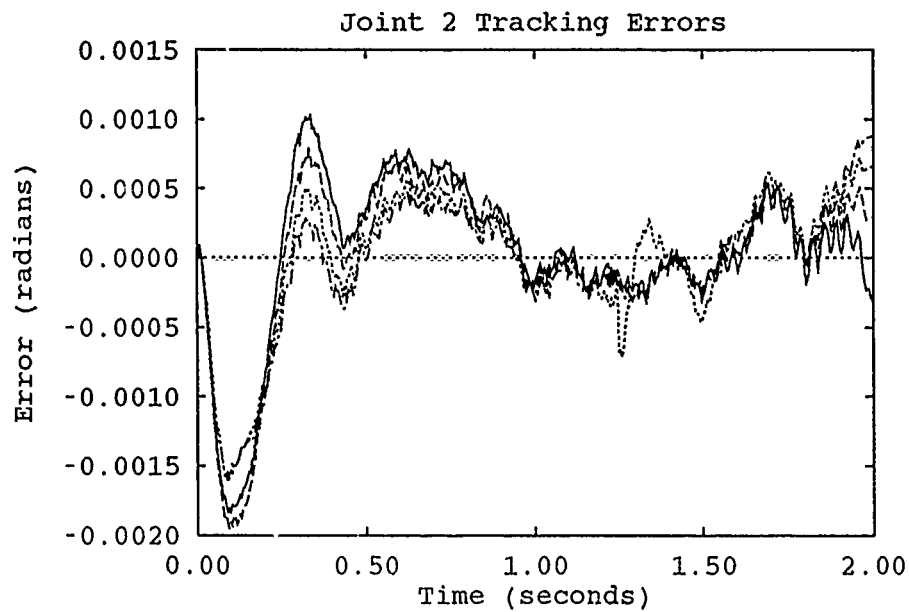


Figure F.5. 13-Parameter Initialized Adaptation Runs - Trajectory 3

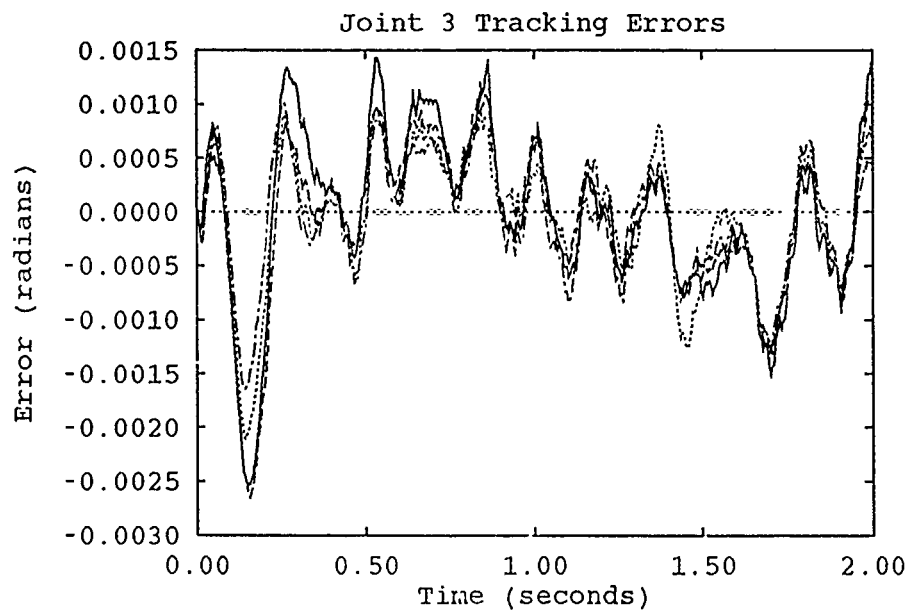


Figure F.6. 13-Parameter Initialized Adaptation Runs - Trajectory 3

—	First Run	Fourth Run
---	Second Run	- - - -	Fifth Run

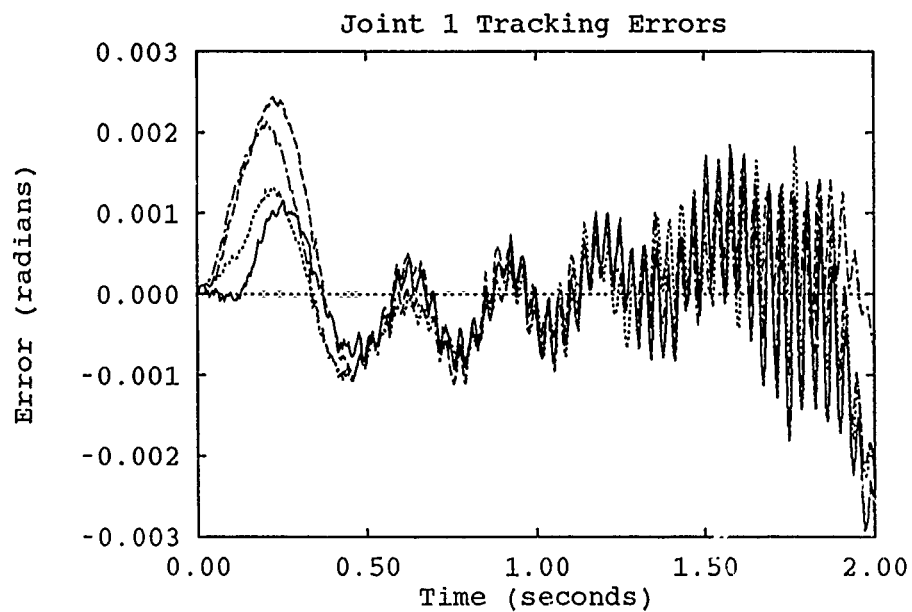


Figure F.7. 13-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

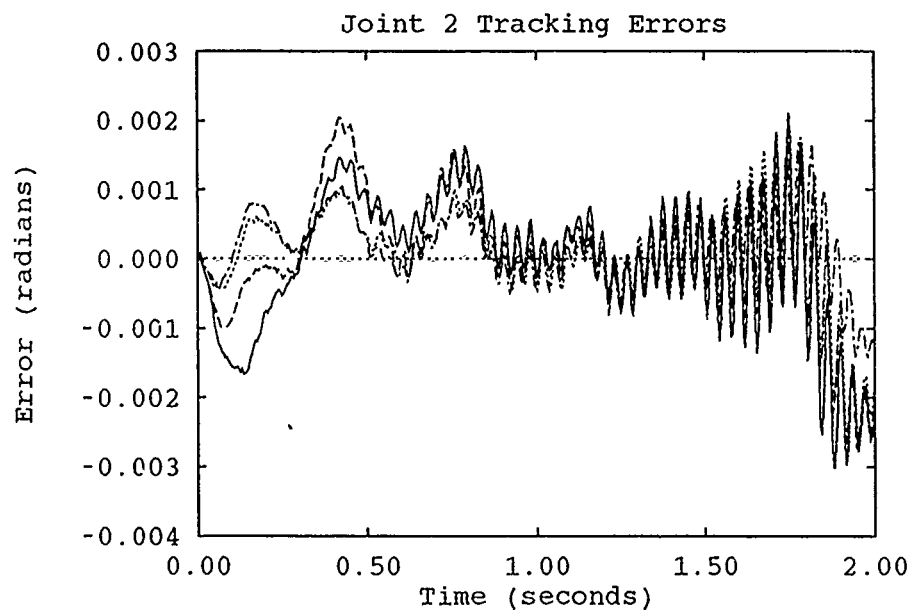


Figure F.8. 13-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

—	First Run	Fourth Run
---	Second Run	- - - -	Fifth Run

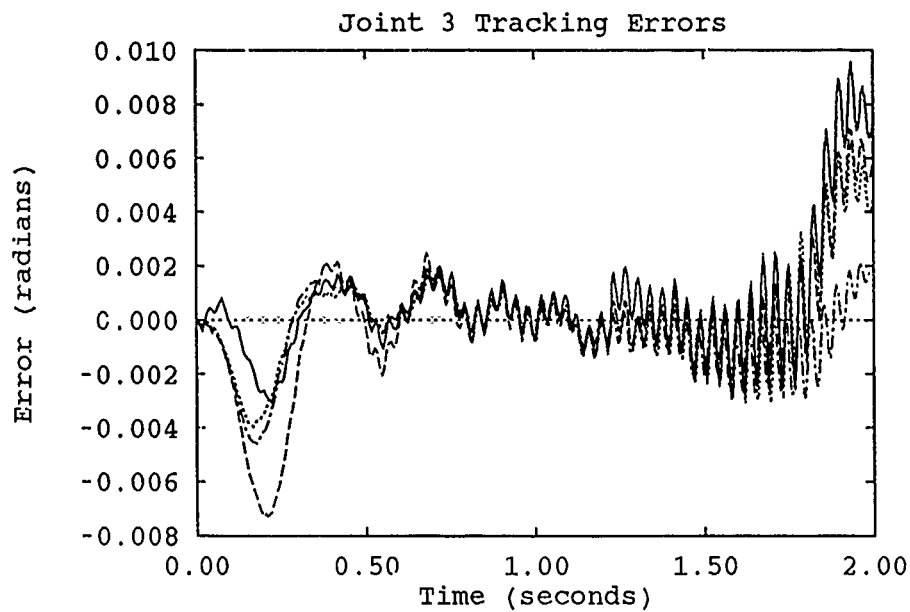


Figure F.9. 13-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

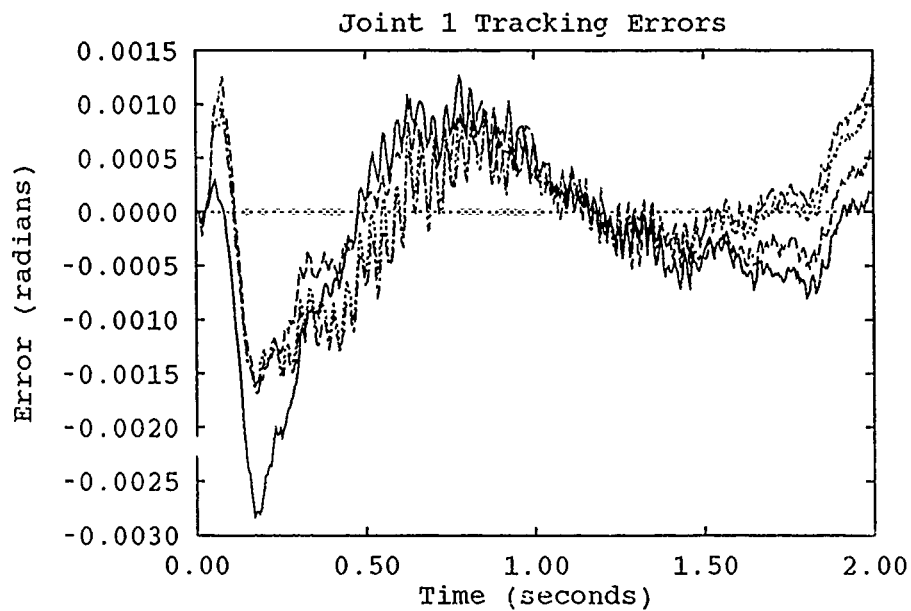


Figure F.10. 13-Parameter Initialized Adaptation Runs - Trajectory 4

—	First Run	Fourth Run
- - -	Second Run	- . - . -	Fifth Run

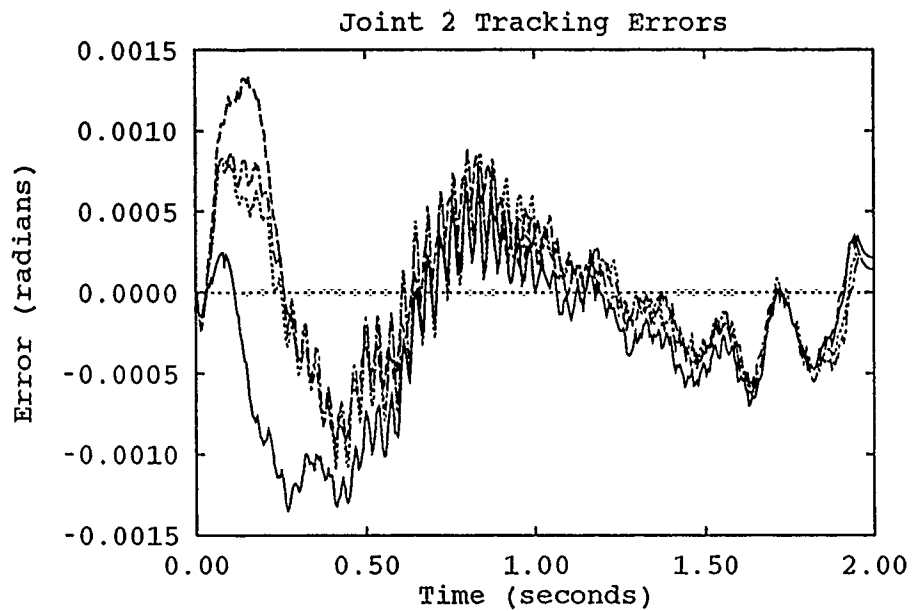


Figure F.11. 13-Parameter Initialized Adaptation Runs - Trajectory 4

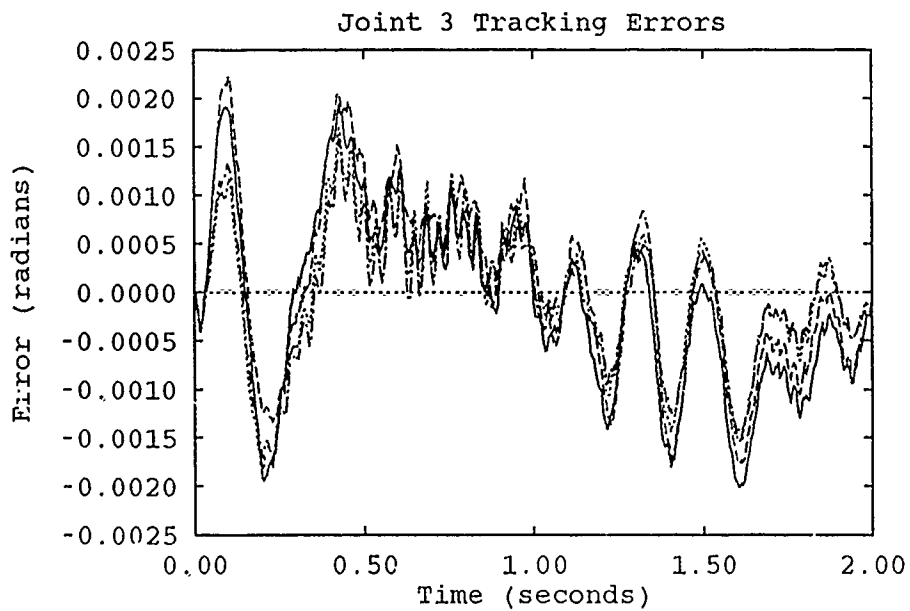


Figure F.12. 13-Parameter Initialized Adaptation Runs - Trajectory 4

—	First Run	Fourth Run
---	Second Run	- - - -	Fifth Run

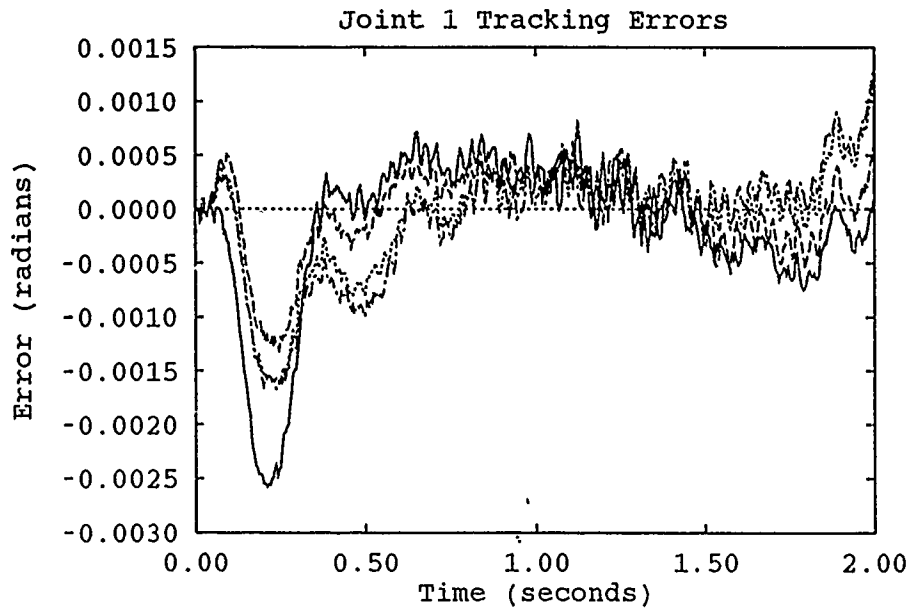


Figure F.13. 13-Parameter Initialized Adaptation Runs - Trajectory 5

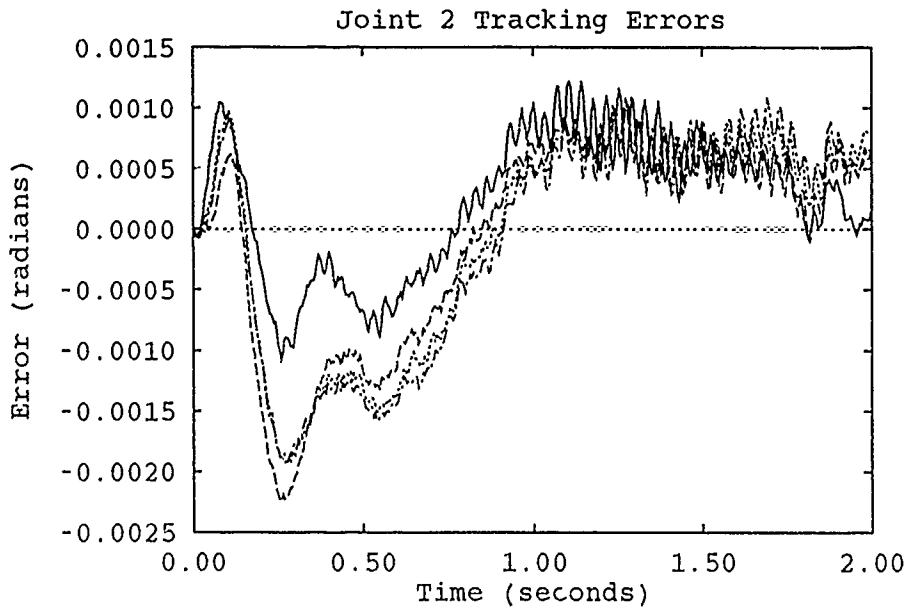


Figure F.14. 13-Parameter Initialized Adaptation Runs - Trajectory 5

—	First Run	Fourth Run
- - -	Second Run	- . - . -	Fifth Run

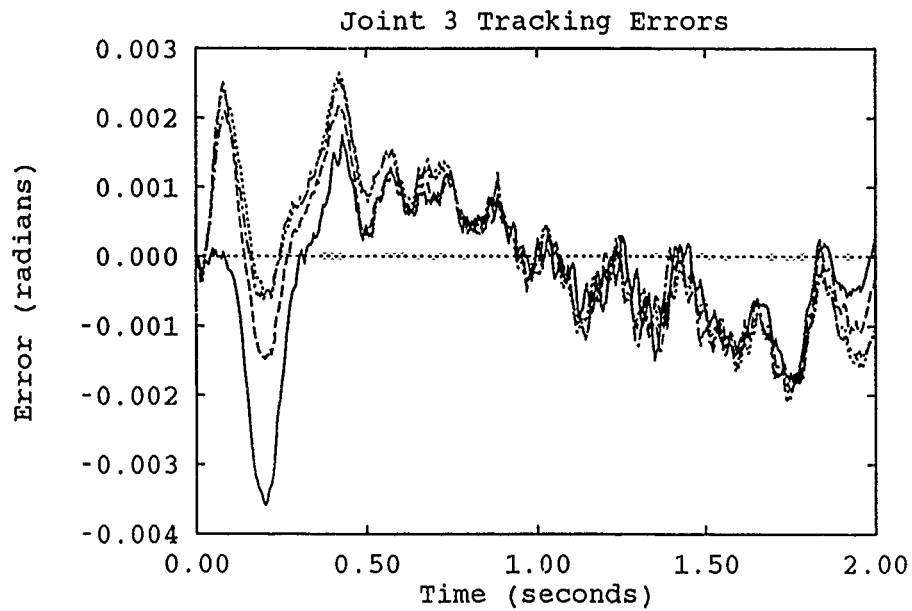


Figure F.15. 13-Parameter Initialized Adaptation Runs - Trajectory 5

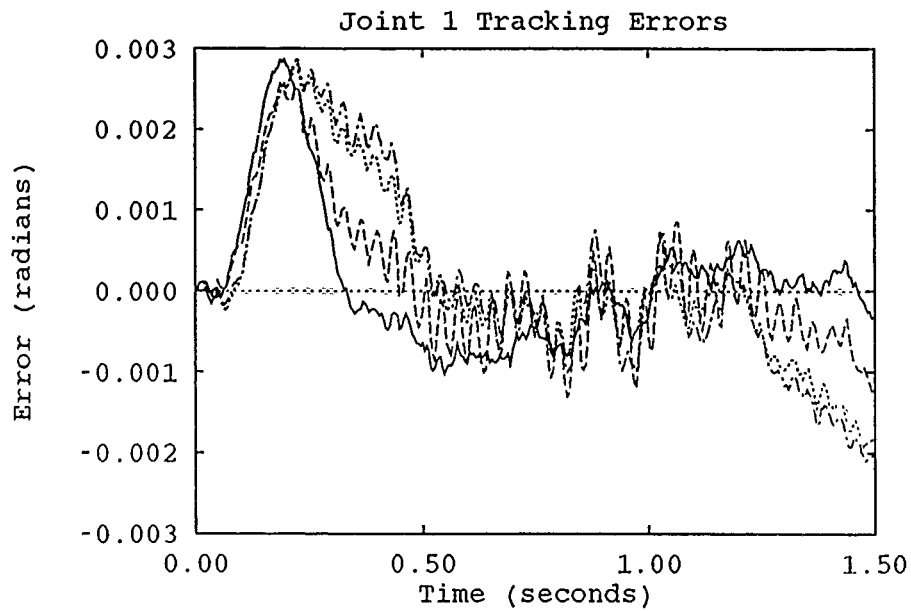


Figure F.16. 13-Parameter Initialized Adaptation Runs - Trajectory 1

—	First Run	Fourth Run
----	Second Run	- - - -	Fifth Run

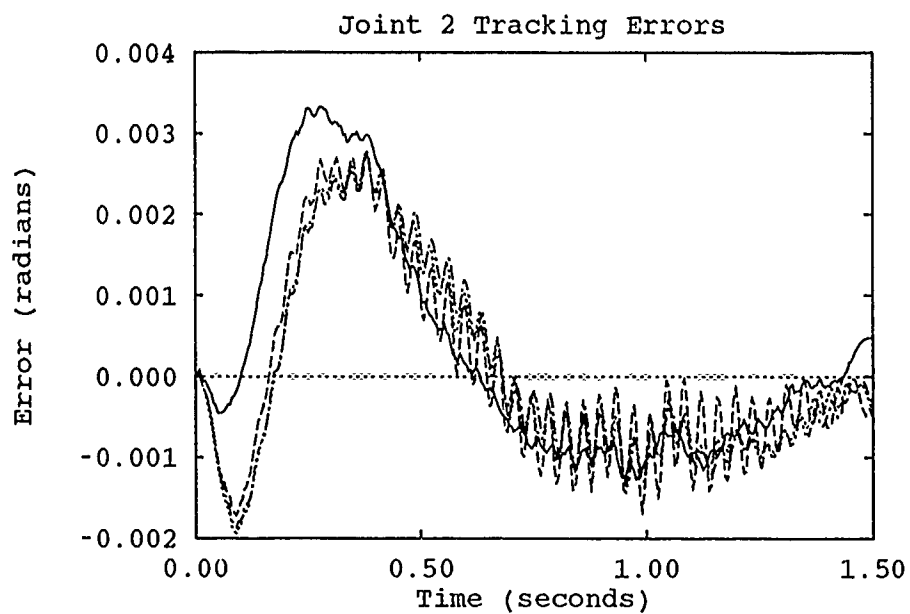


Figure F.17. 13-Parameter Initialized Adaptation Runs - Trajectory 1

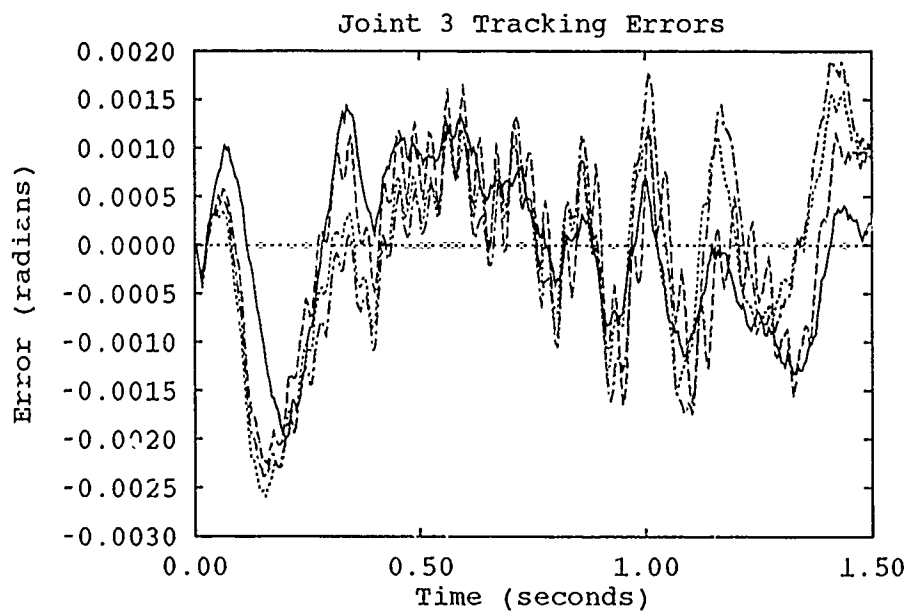


Figure F.18. 13-Parameter Initialized Adaptation Runs - Trajectory 1

—	First Run	Fourth Run
- - -	Second Run	- . - .	Fifth Run

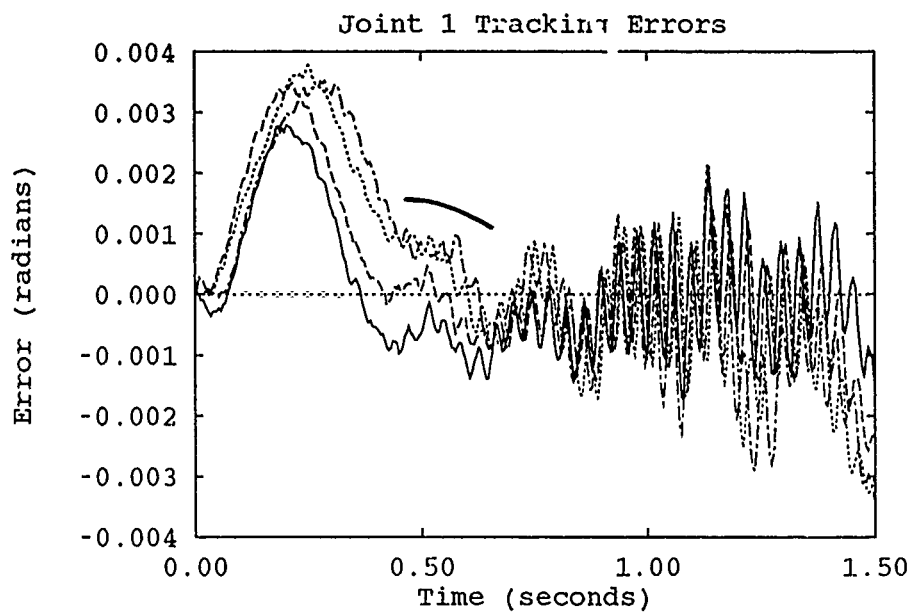


Figure F.19. 13-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

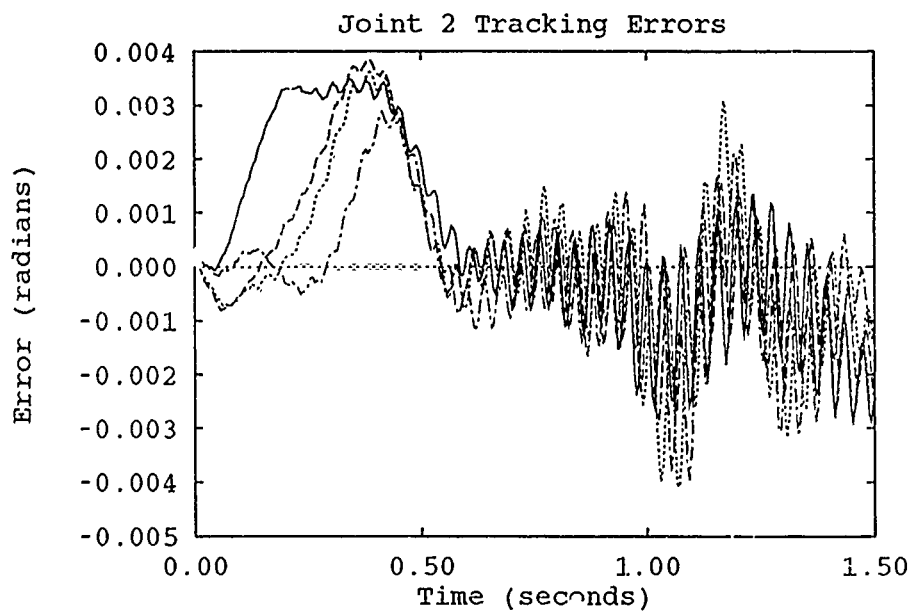


Figure F.20. 13 Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

—	First Run	Fourth Run
---	Second Run	- - - -	Fifth Run

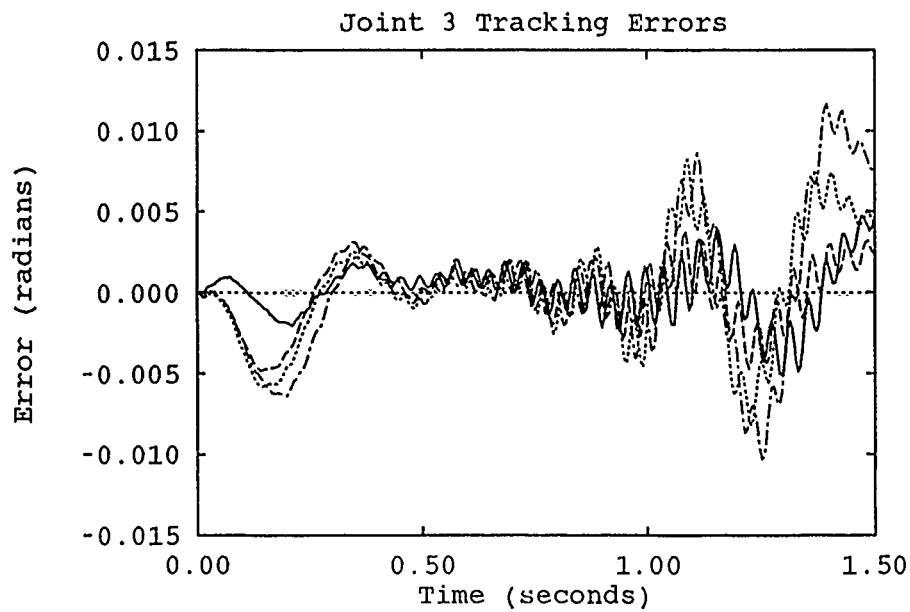


Figure F.21. 13-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

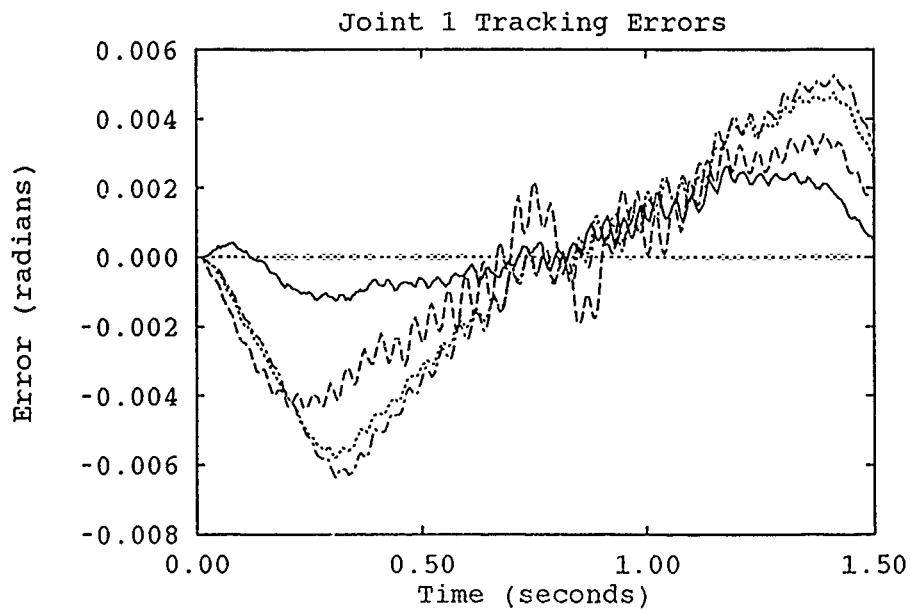


Figure F.22. 13-Parameter Initialized Adaptation Runs - Trajectory 6

—	First Run	· · · · ·	Fourth Run
- - - - -	Second Run	- · - · -	Fifth Run

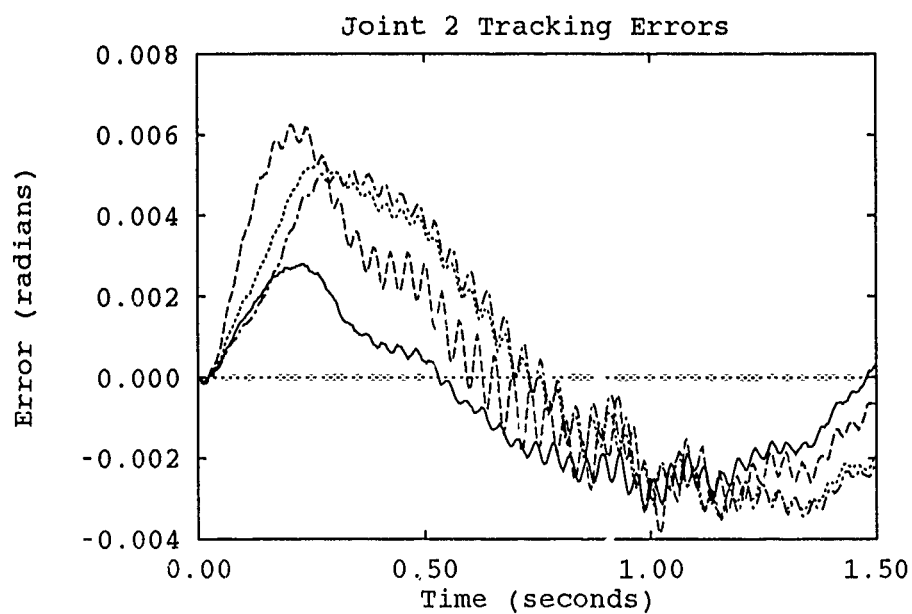


Figure F.23. 13-Parameter Initialized Adaptation Runs - Trajectory 6

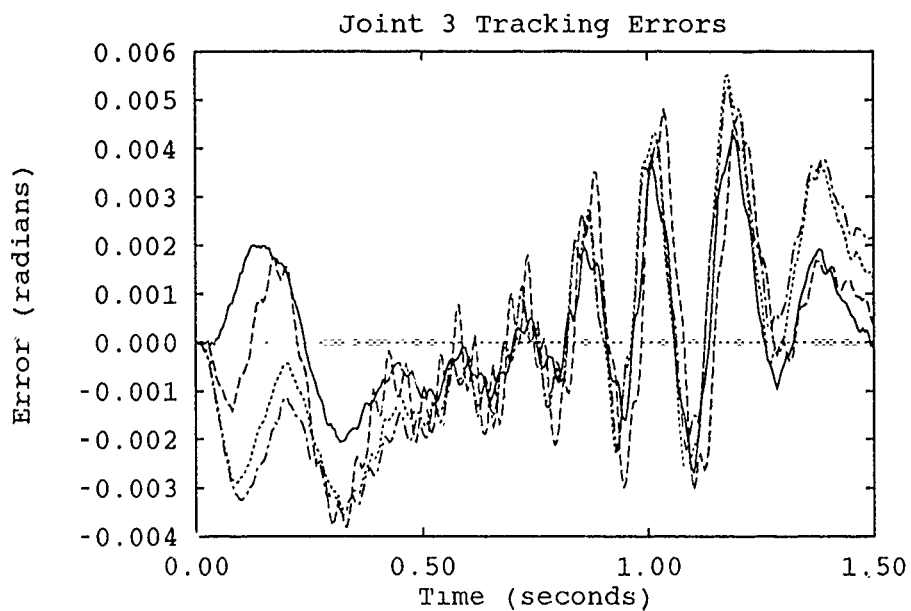


Figure F.24. 13-Parameter Initialized Adaptation Runs - Trajectory 6

—	First Run	Fourth Run
---	Second Run	- - - -	Fifth Run

Appendix G. *16-Parameter Initialized Learning Runs*

This section contain the results of 16-parameter adaptation runs for each test trajectory using initialized parameters. Each trajectory was run five times to allow the controller to adapt.

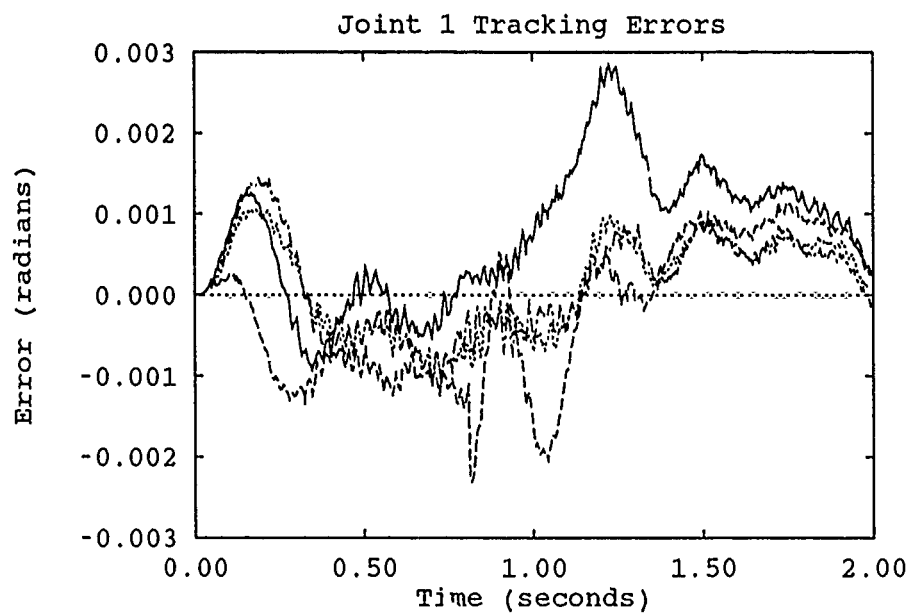


Figure G.1. 16-Parameter Initialized Adaptation Runs - Trajectory 2

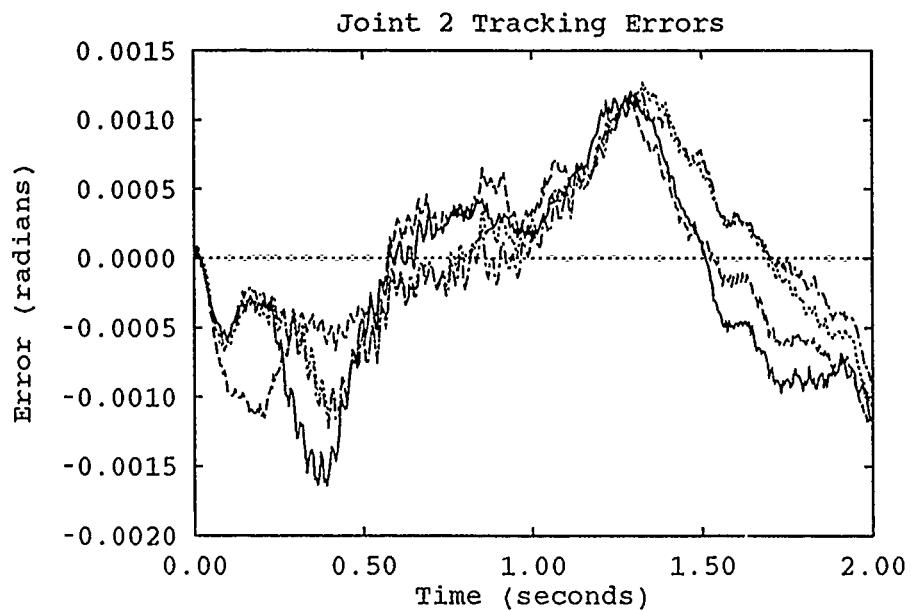


Figure G.2. 16-Parameter Initialized Adaptation Runs - Trajectory 2

—	First Run	Fourth Run
- - - -	Second Run	- · - · -	Fifth Run

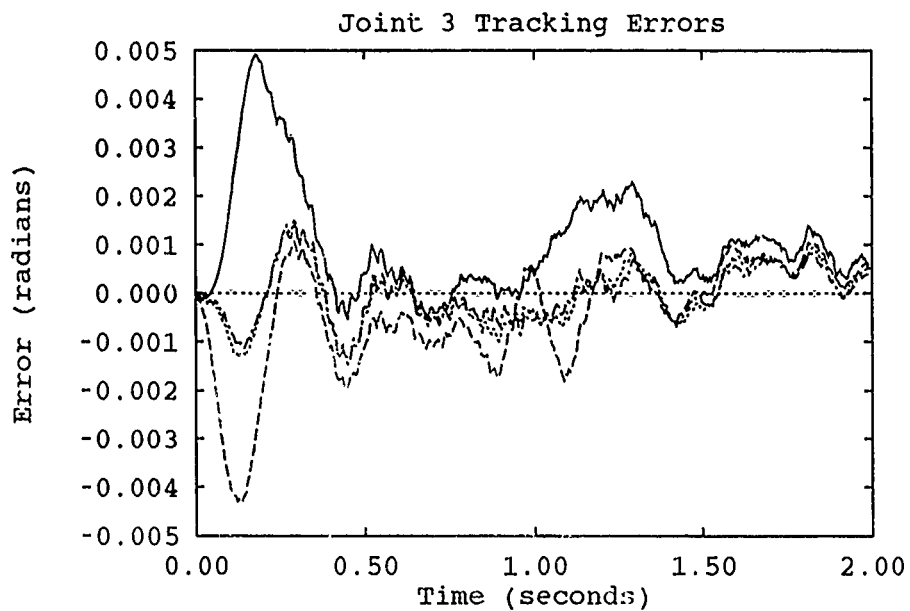


Figure G.3. 16-Parameter Initialized Adaptation Runs - Trajectory 2

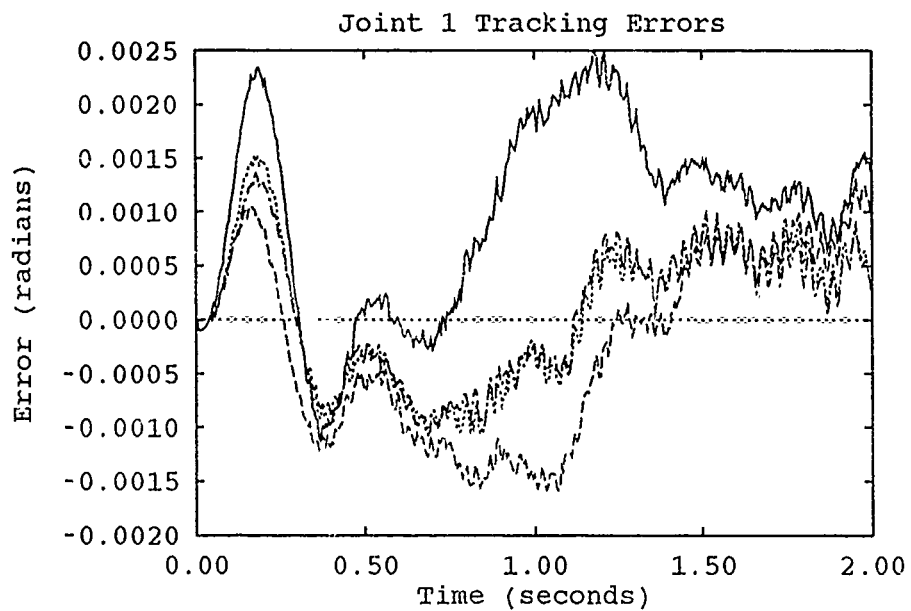


Figure G.4. 16-Parameter Initialized Adaotation Runs - Trajectory 3

—	First Run	Fourth Run
- - -	Second Run	- . - . -	Fifth Run

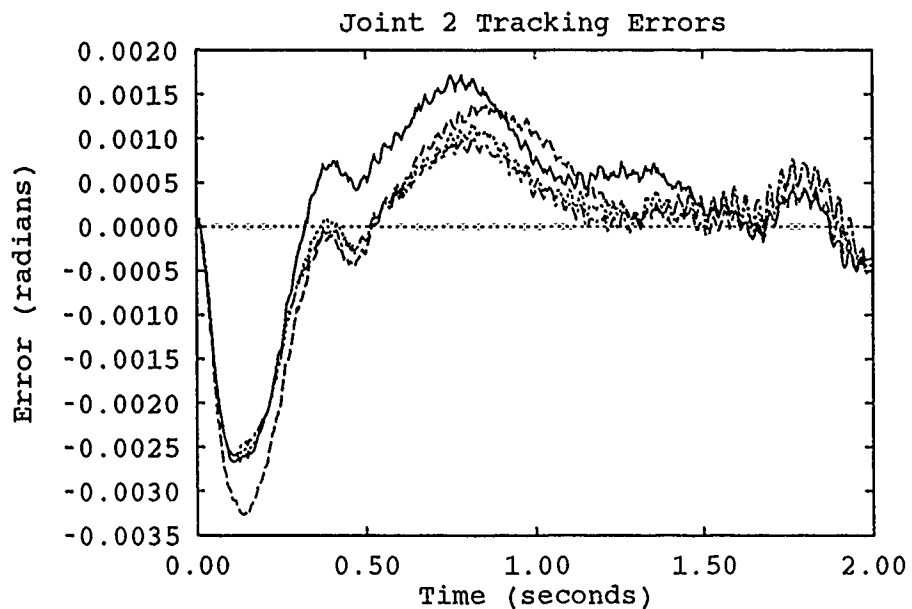


Figure G.5. 16-Parameter Initialized Adaptation Runs - Trajectory 3

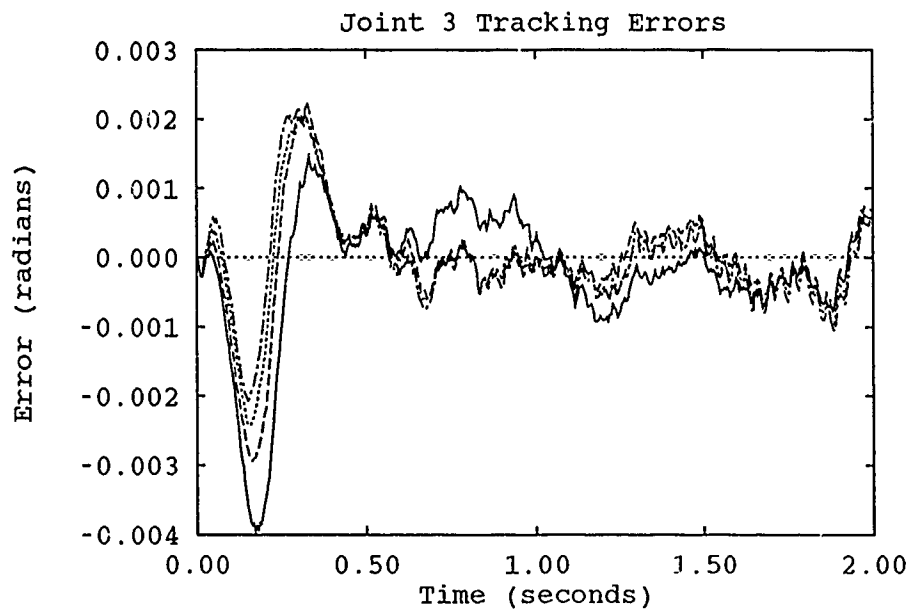


Figure G.6. 16-Parameter Initialized Adaptation Runs - Trajectory 3

—	First Run	Fourth Run
- - -	Second Run	- . - . -	Fifth Run

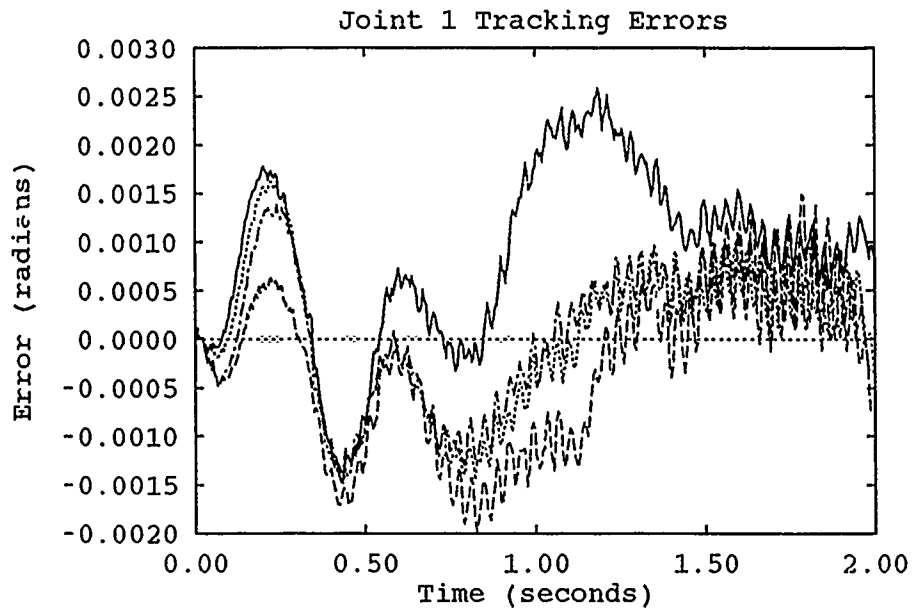


Figure G.7. 16-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

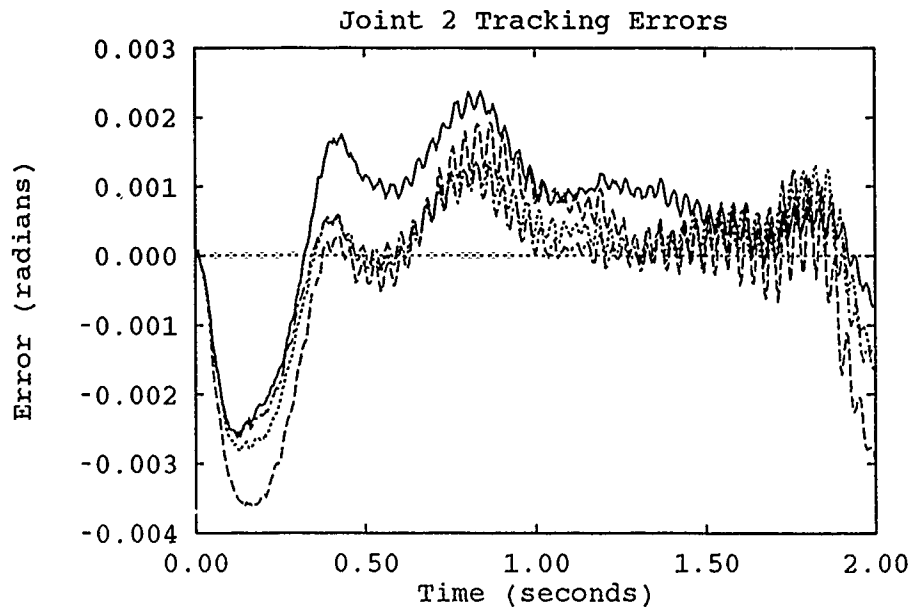


Figure G.8. 16-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

—	First Run	Fourth Run
- - - -	Second Run	- . - . -	Fifth Run

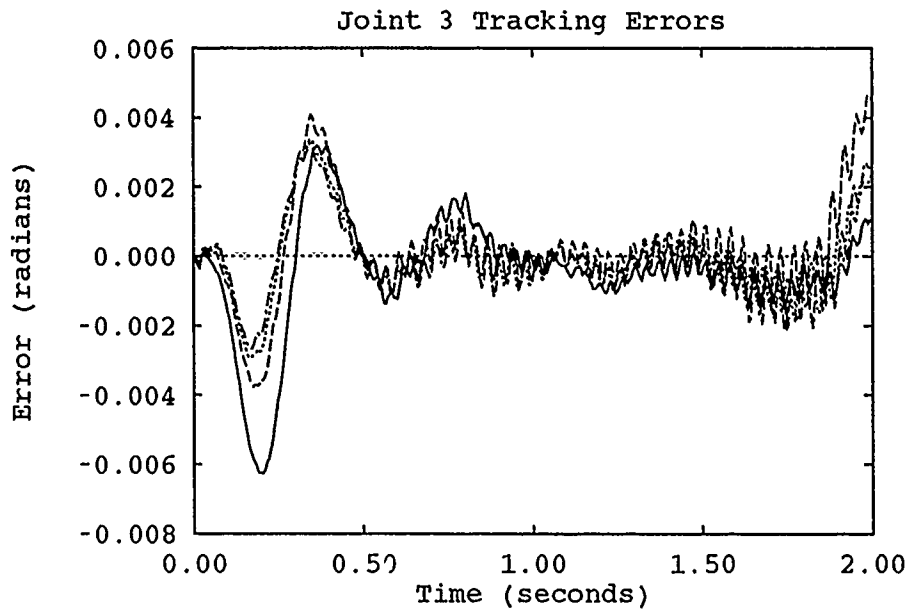


Figure G.9. 16-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

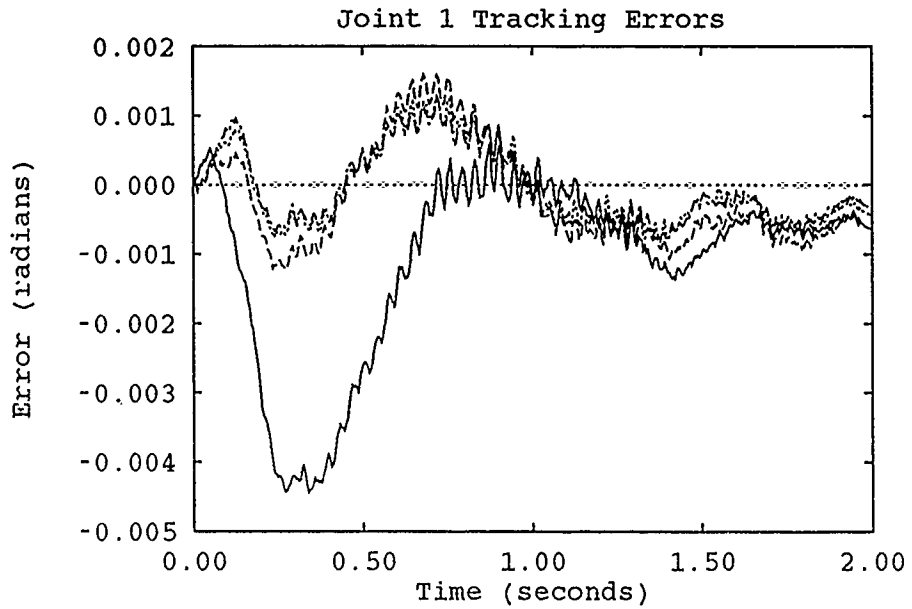


Figure G.10. 16-Parameter Initialized Adaptation Runs - Trajectory 4

—	First Run	Fourth Run
- - -	Second Run	- . - .	Fifth Run

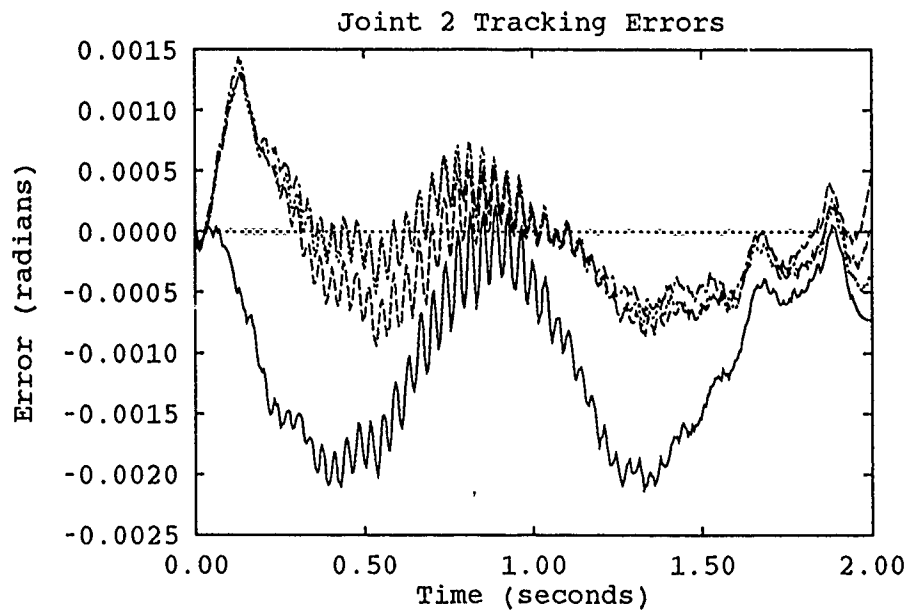


Figure G.11. 16-Parameter Initialized Adaptation Runs - Trajectory 4

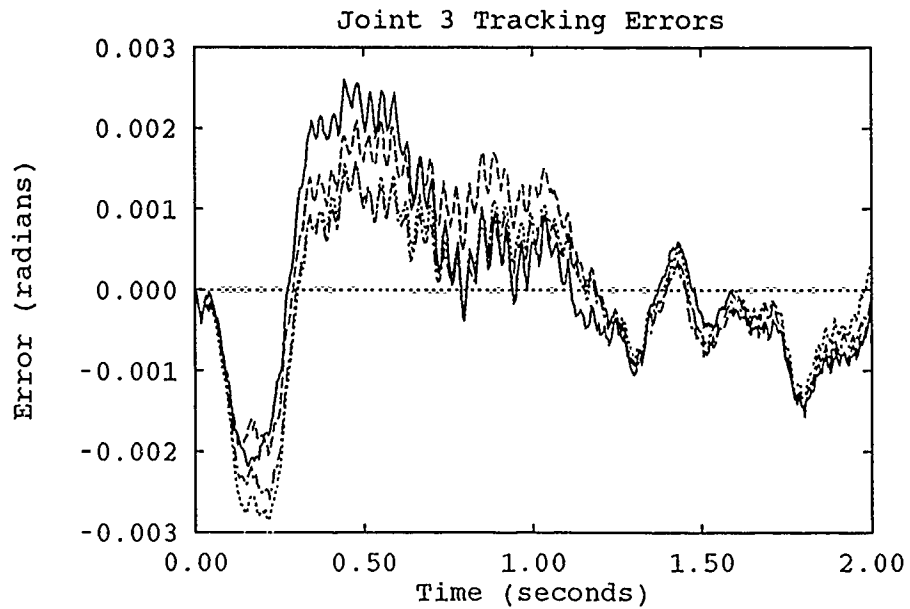


Figure G.12. 16-Parameter Initialized Adaptation Runs - Trajectory 4

—	First Run	Fourth Run
- - -	Second Run	- . - .	Fifth Run

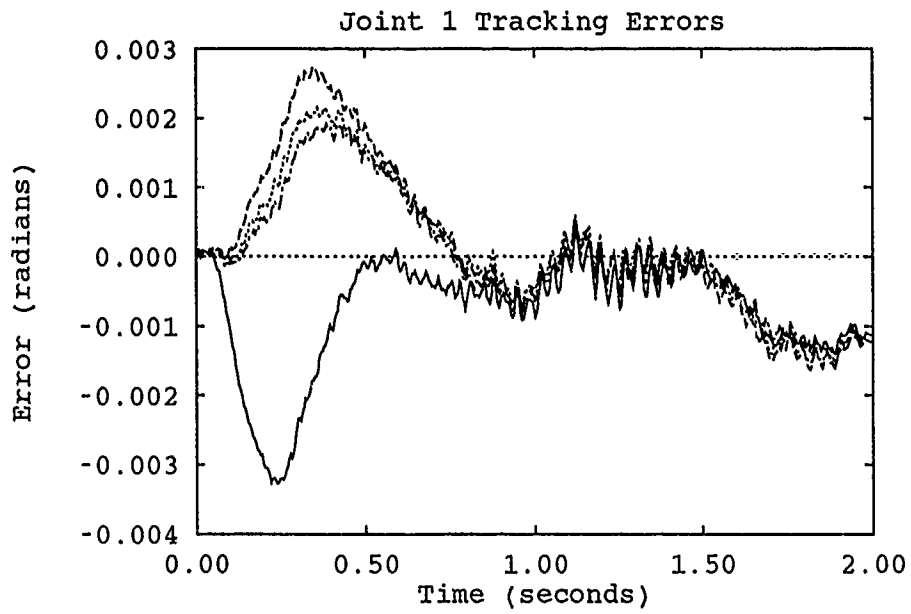


Figure G.13. 16-Parameter Initialized Adaptation Runs - Trajectory 5

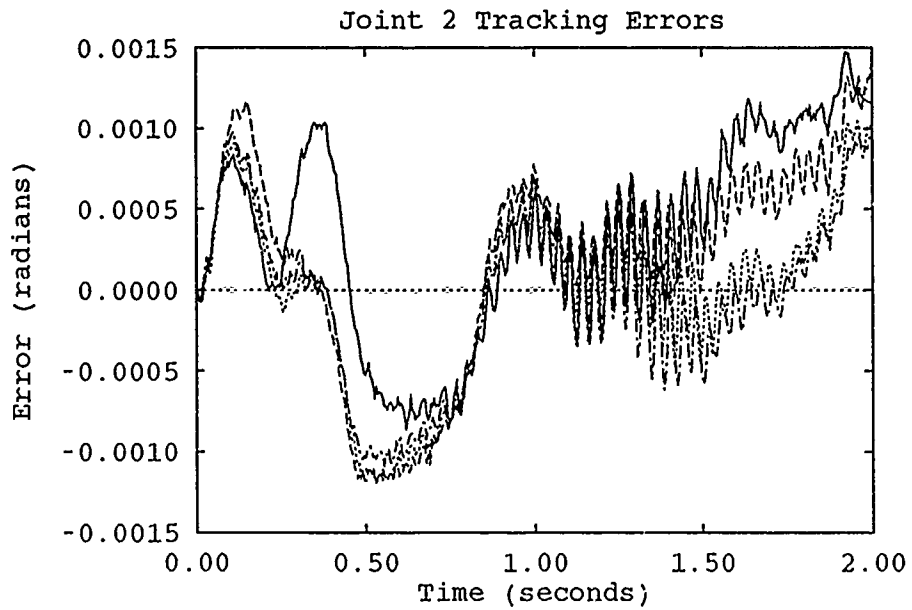


Figure G.14. 16-Parameter Initialized Adaptation Runs - Trajectory 5

—	First Run	Fourth Run
- - - -	Second Run	- · - · -	Fifth Run

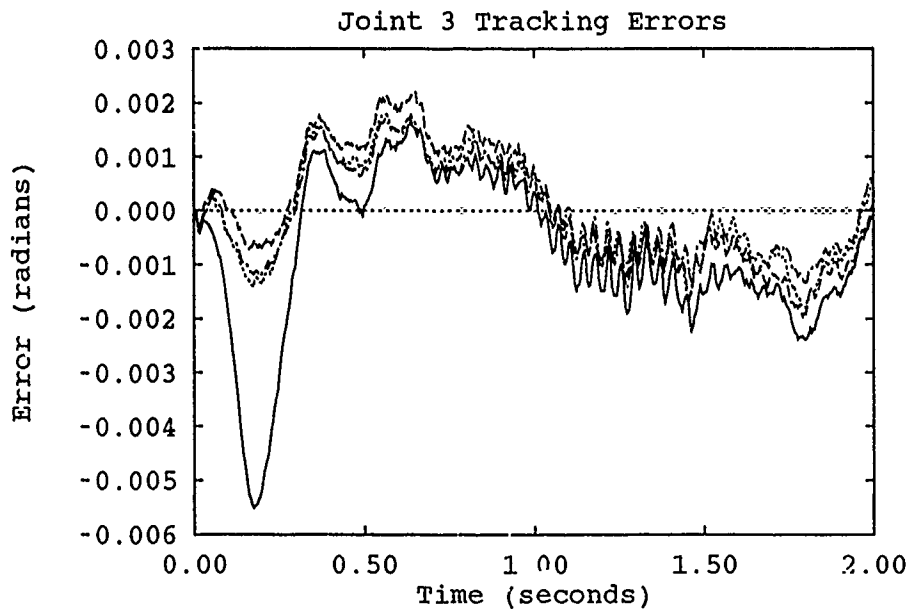


Figure G.15. 16-Parameter Initialized Adaptation Runs - Trajectory 5

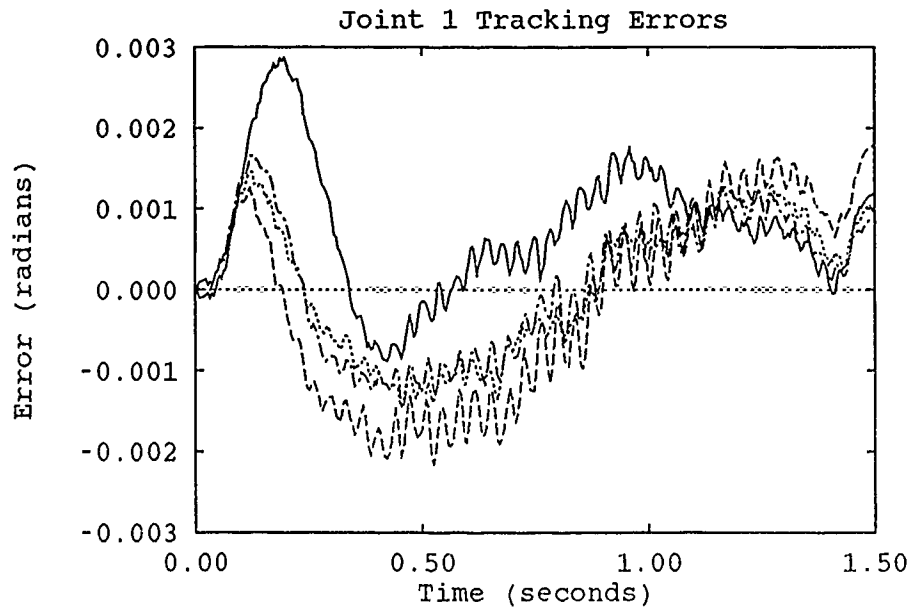


Figure G.16. 16-Parameter Initialized Adaptation Runs - Trajectory 1

—	First Run	Fourth Run
----	Second Run	-.-.-.-	Fifth Run

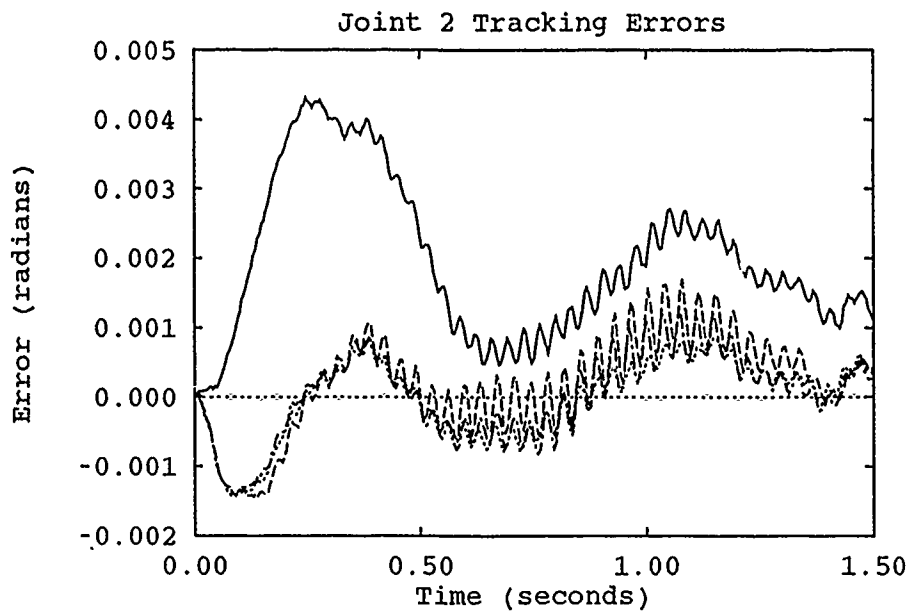


Figure G.17. 16-Parameter Initialized Adaptation Runs - Trajectory 1

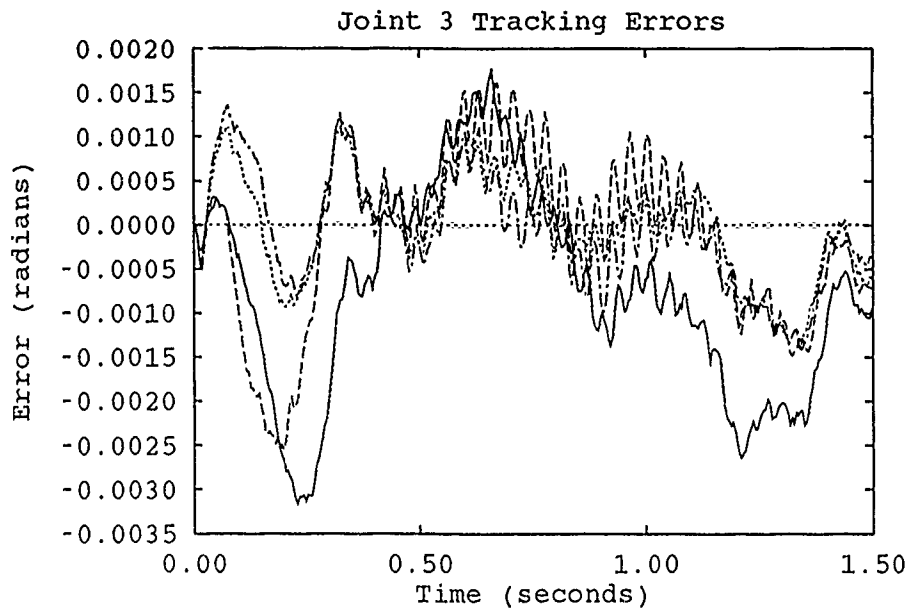


Figure G.18. 16-Parameter Initialized Adaptation Runs - Trajectory 1

—	First Run	Fourth Run
- - - -	Second Run	- · - · -	Fifth Run

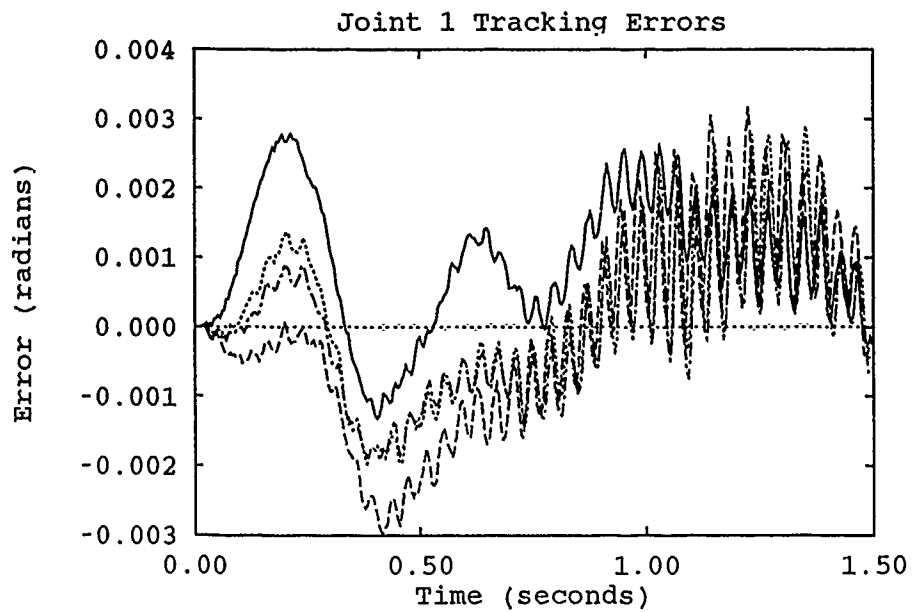


Figure G.19. 16-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

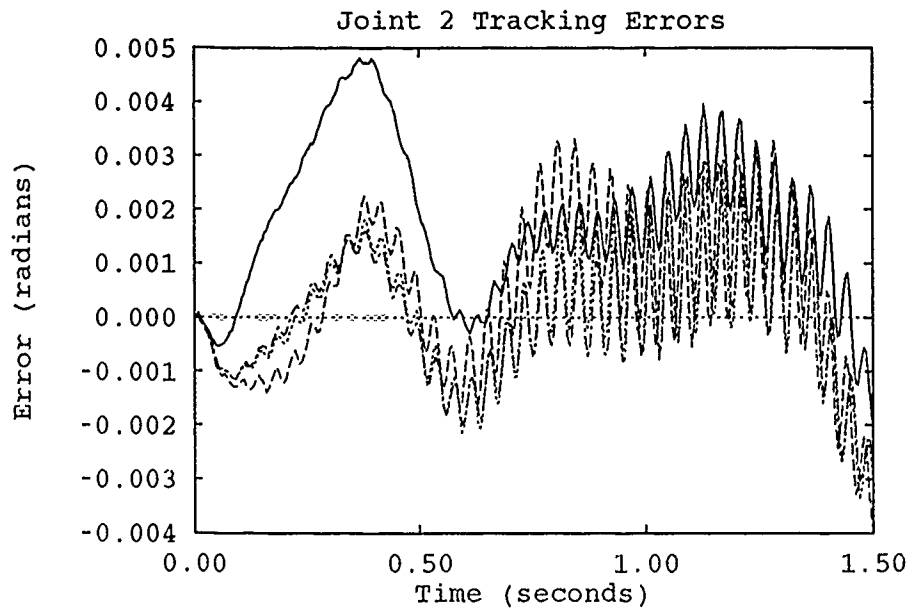


Figure G.20. 16-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

—	First Run	Fourth Run
----	Second Run	-.-.-	Fifth Run

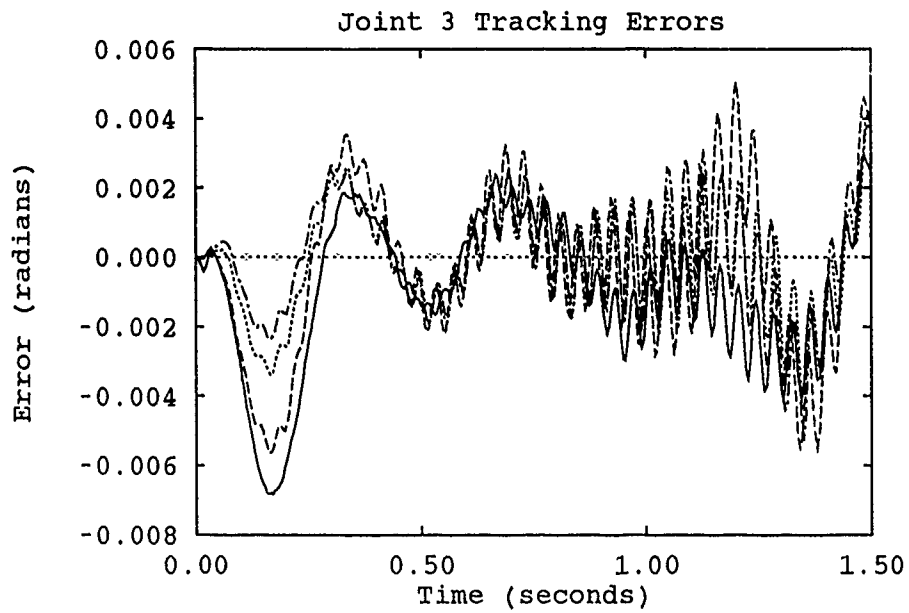


Figure G.21. 16-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

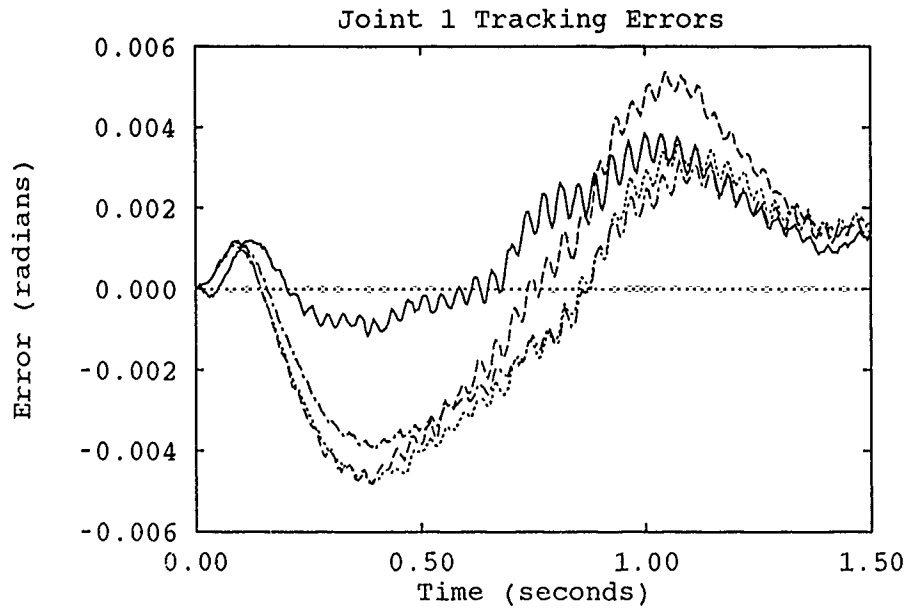


Figure G.22. 16-Parameter Initialized Adaptation Runs - Trajectory 6

—	First Run	Fourth Run
- - - -	Second Run	- . - . -	Fifth Run

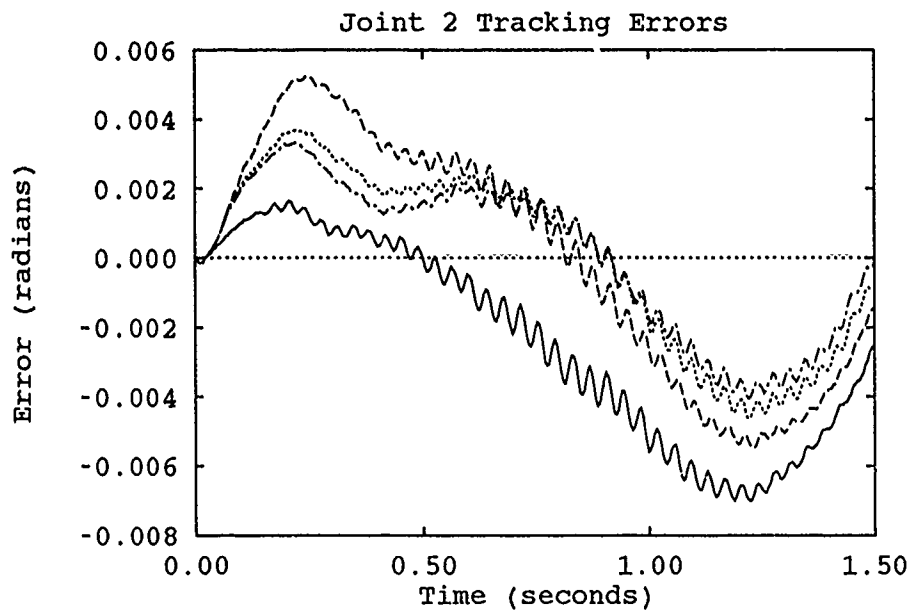


Figure G.23. 16-Parameter Initialized Adaptation Runs - Trajectory 6

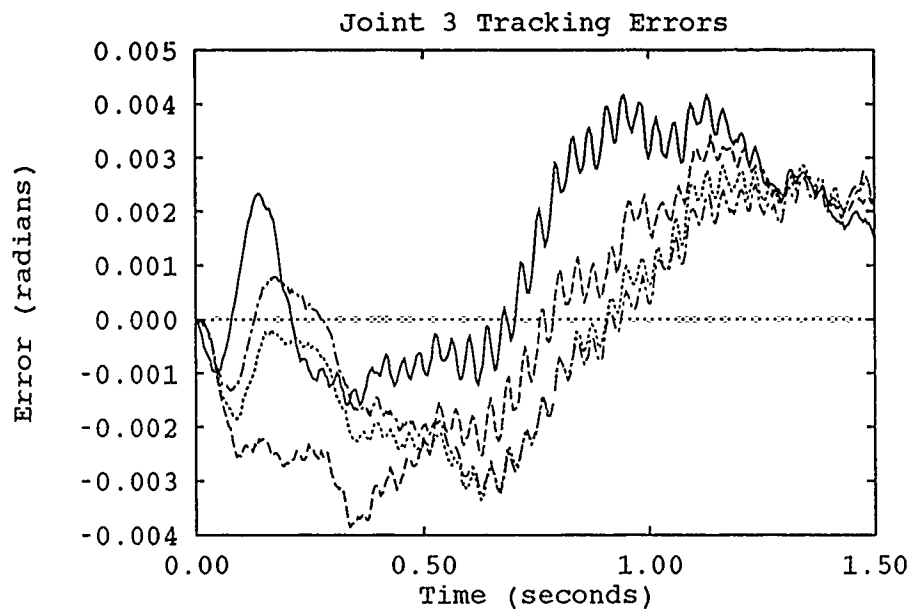


Figure G.24. 16-Parameter Initialized Adaptation Runs - Trajectory 6

—	First Run	Fourth Run
- - - -	Second Run	- . - . -	Fifth Run

Appendix H. *19-Parameter Initialized Learning Runs*

This section contain the results of 19-parameter adaptation runs for each test trajectory using initialized parameters. Each trajectory was run five times to allow the controller to adapt.

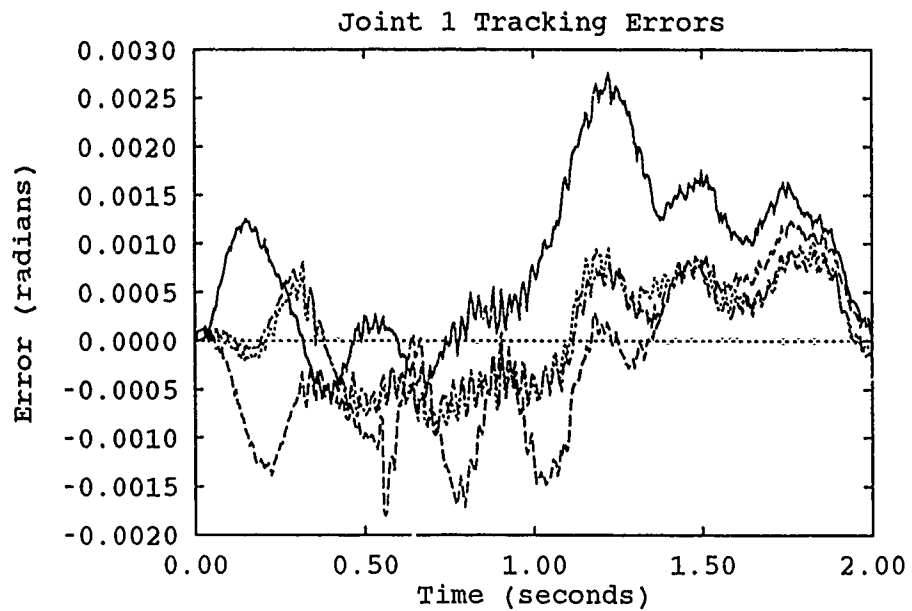


Figure H.1. 19-Parameter Initialized Adaptation Runs - Trajectory 2

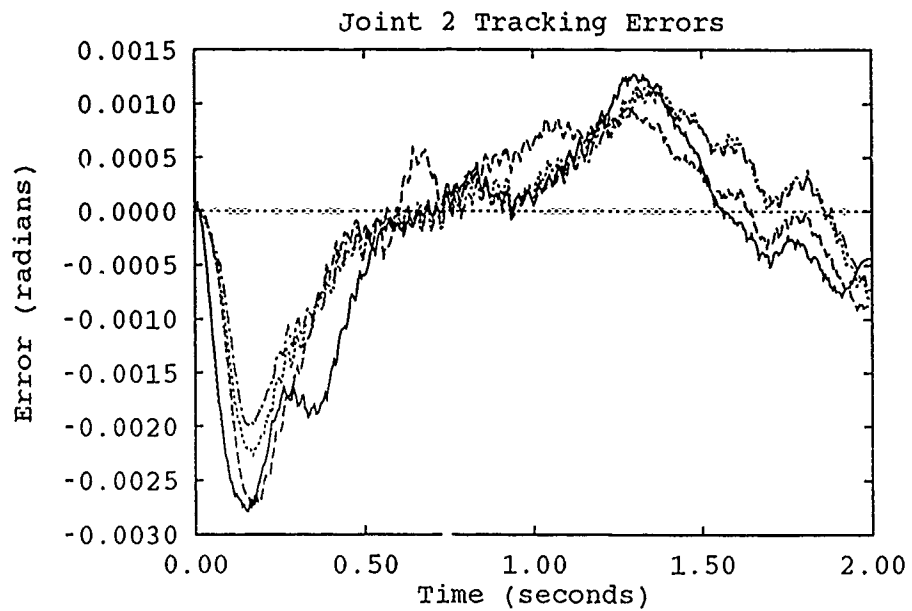


Figure H.2. 19-Parameter Initialized Adaptation Runs - Trajectory 2

—	First Run	Fourth Run
- - -	Second Run	- . - .	Fifth Run

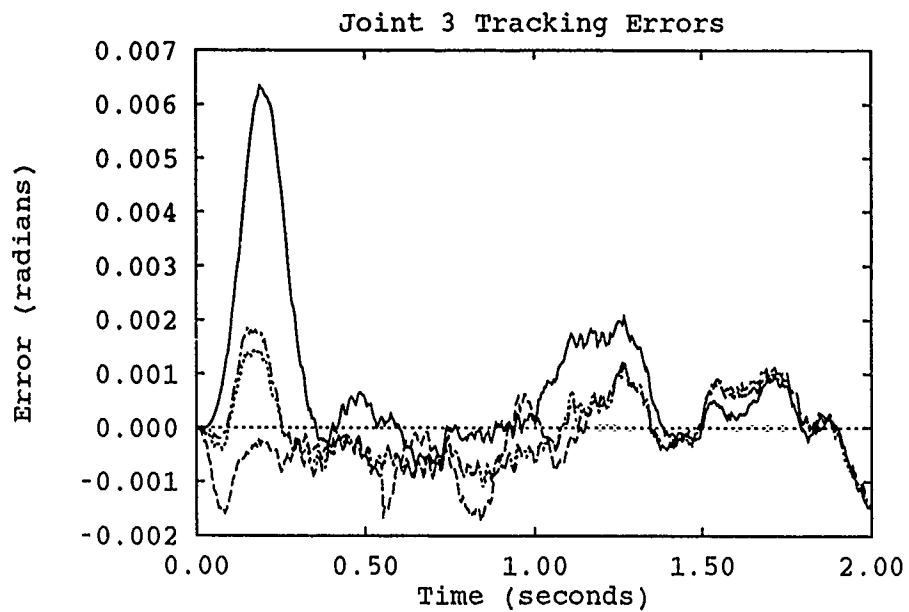


Figure H.3. 19-Parameter Initialized Adaptation Runs - Trajectory 2

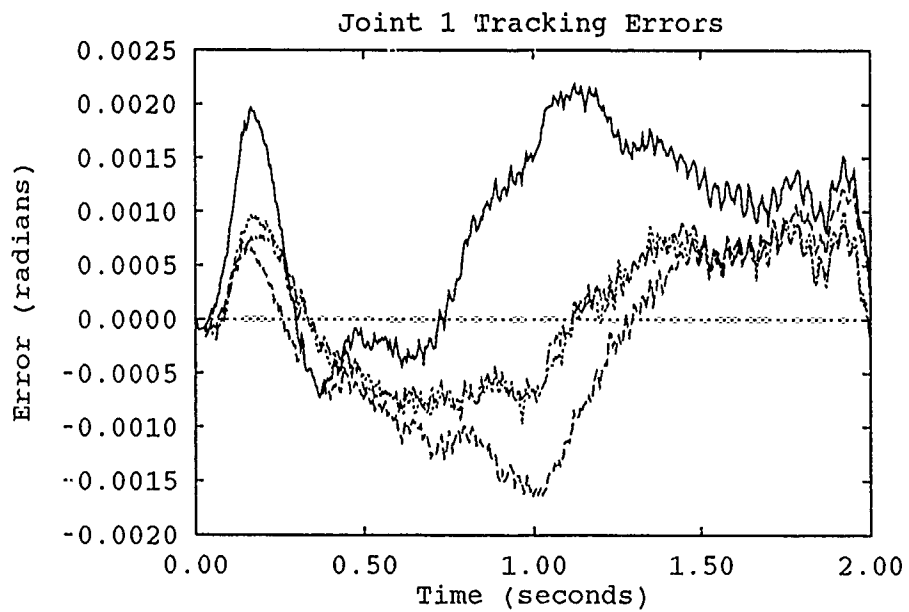


Figure H.4. 19-Parameter Initialized Adaptation Runs - Trajectory 3

—	First Run	Fourth Run
- - -	Second Run	- · - · -	Fifth Run

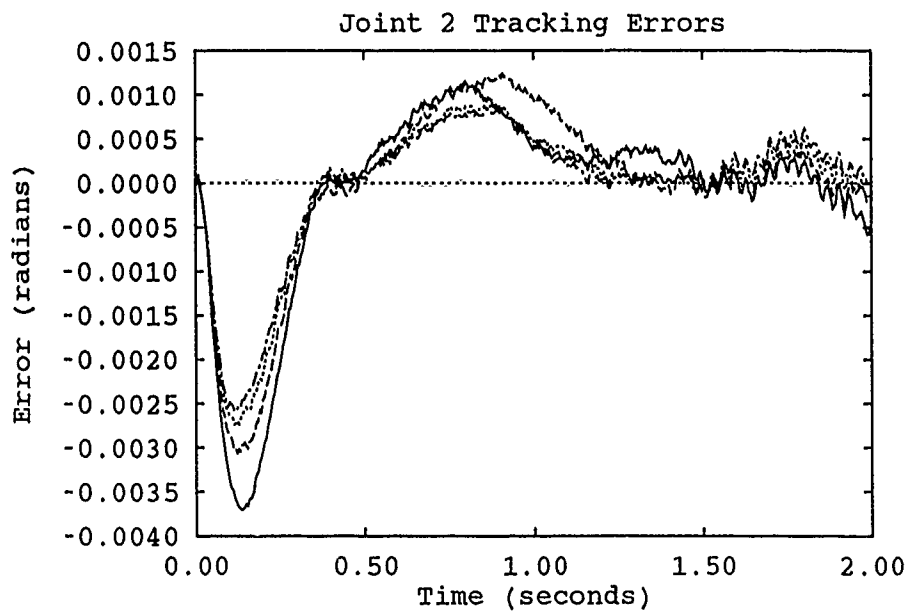


Figure H.5. 19-Parameter Initialized Adaptation Runs - Trajectory 3

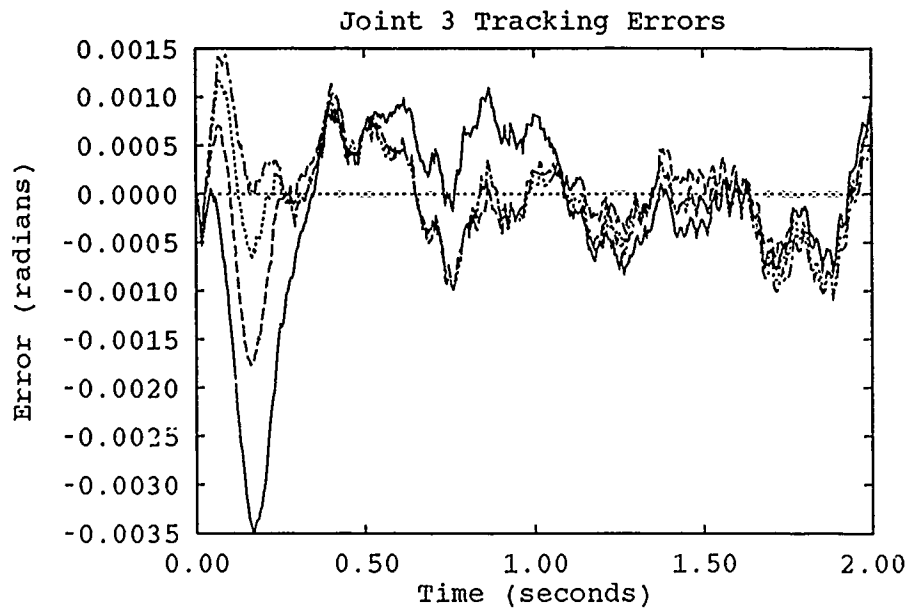


Figure H.6. 19-Parameter Initialized Adaptation Runs - Trajectory 3

—	First Run	Fourth Run
----	Second Run	-.-.-.-	Fifth Run

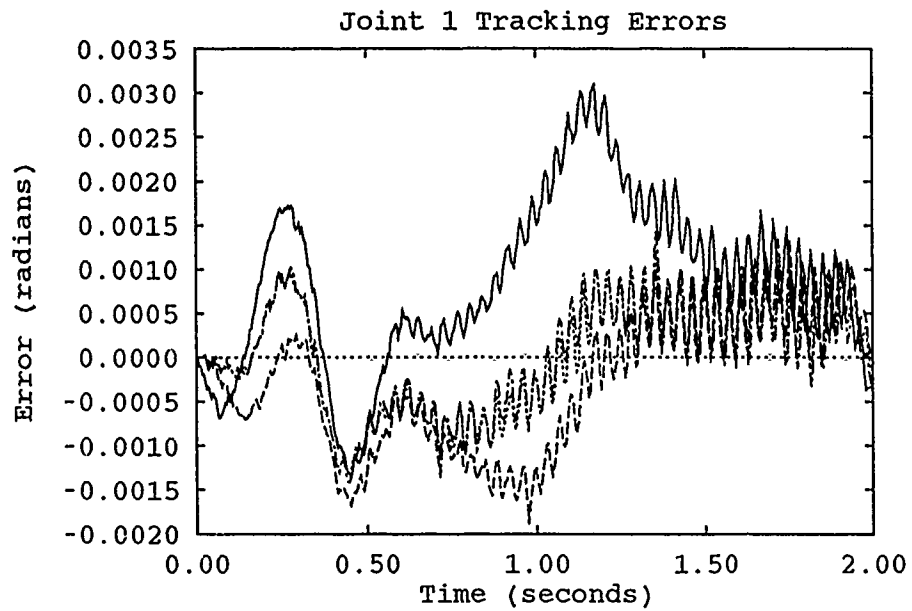


Figure H.7. 19-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

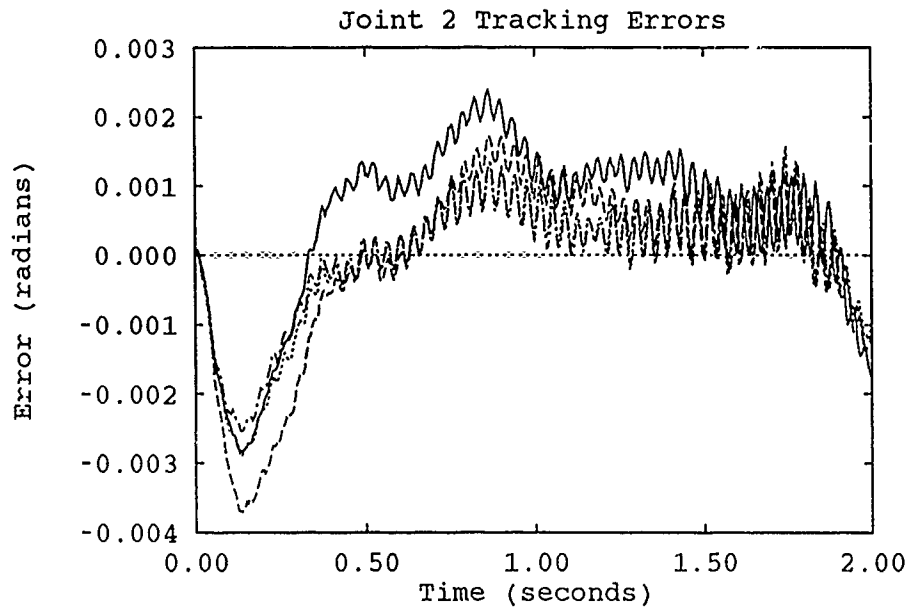


Figure H.8. 19-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

—	First Run	Fourth Run
- - -	Second Run	- . - . -	Fifth Run

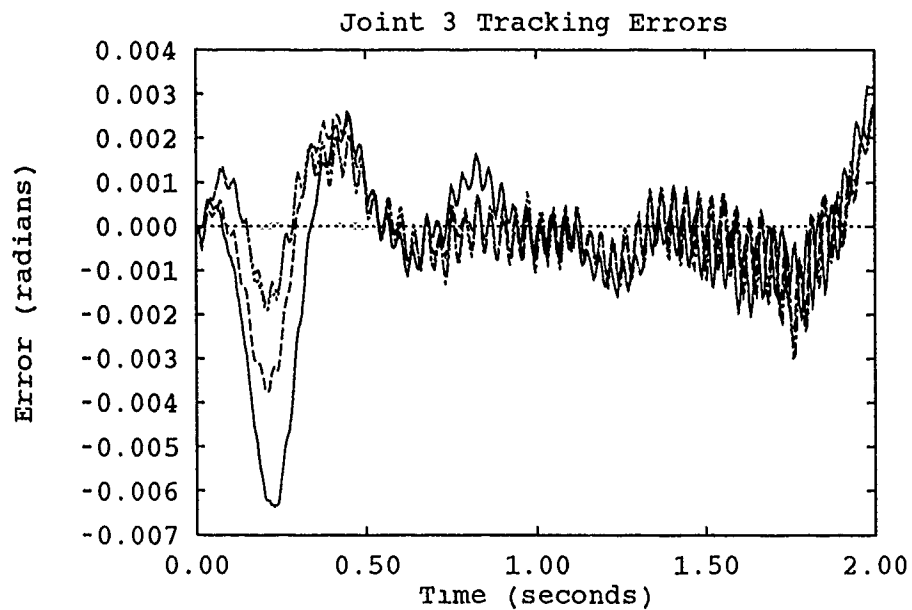


Figure H.9. 19-Parameter Initialized Adaptation Runs - Trajectory 3 w/ Payload

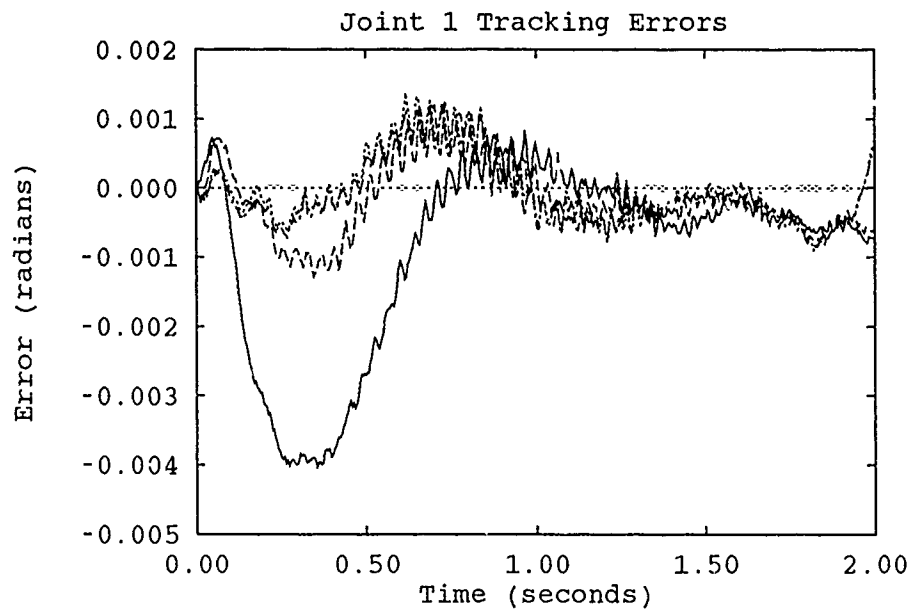


Figure H.10. 19-Parameter Initialized Adaptation Runs - Trajectory 4

—	First Run	Fourth Run
----	Second Run	-.-.-.	Fifth Run

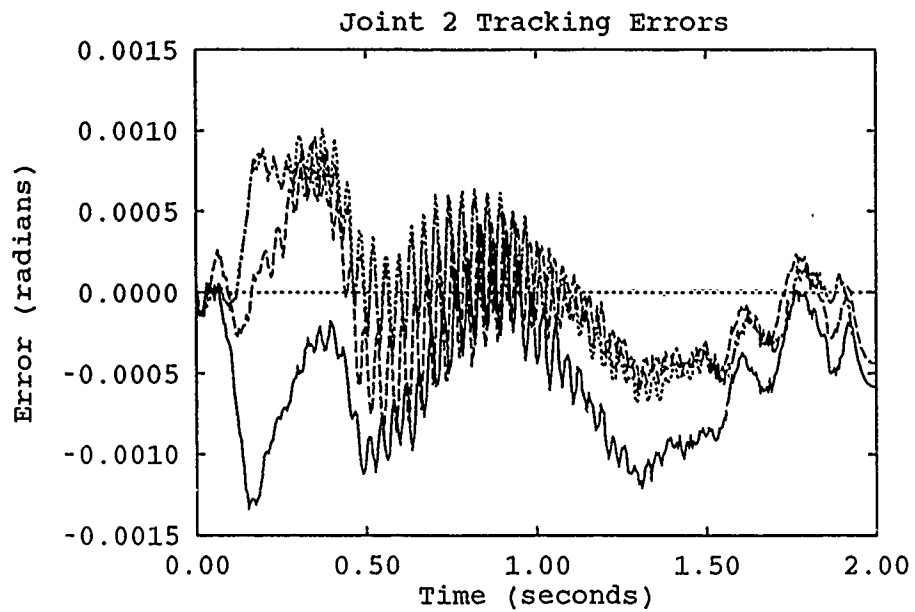


Figure H.11. 19-Parameter Initialized Adaptation Runs - Trajectory 4

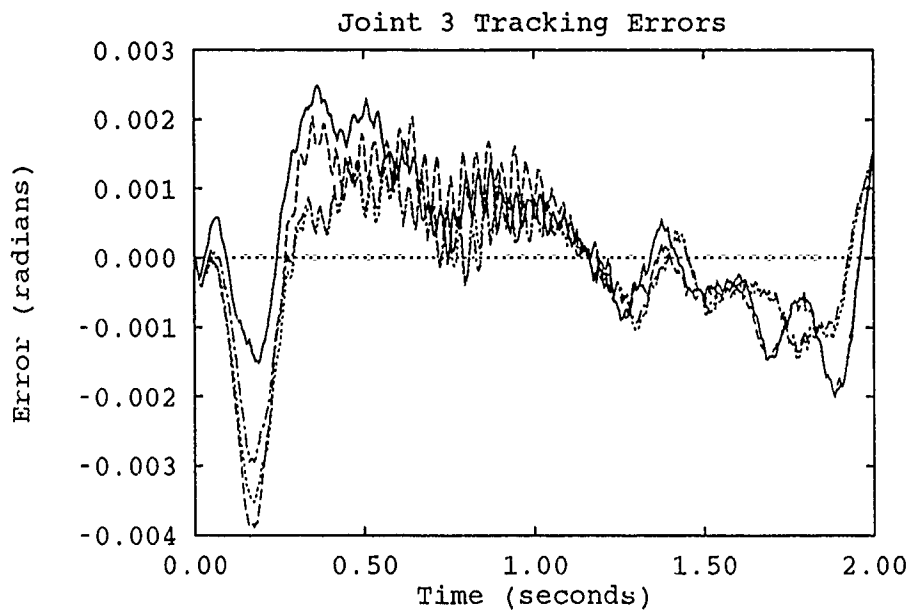


Figure H.12. 19-Parameter Initialized Adaptation Runs - Trajectory 4

—	First Run	Fourth Run
----	Second Run	- - - -	Fifth Run

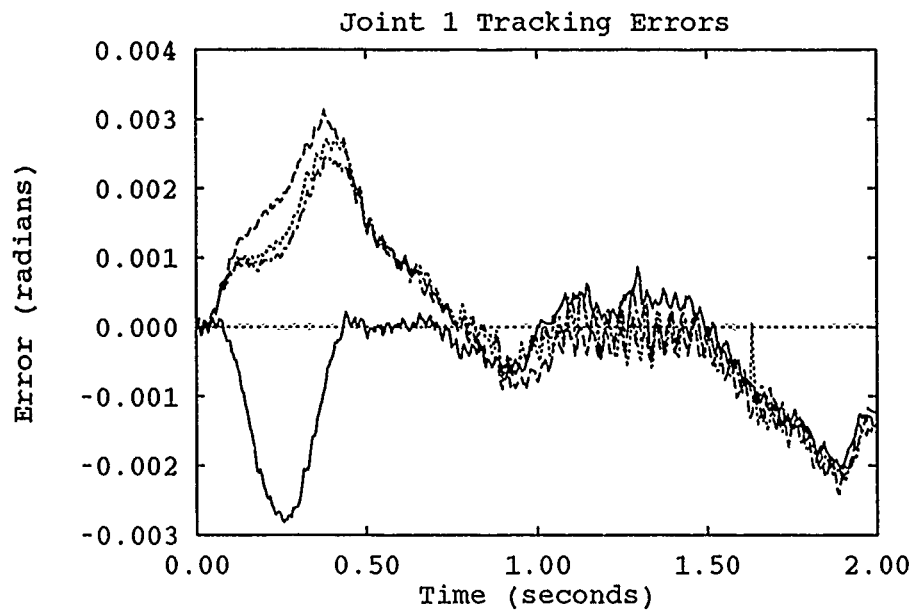


Figure H.13. 19-Parameter Initialized Adaptation Runs - Trajectory 5

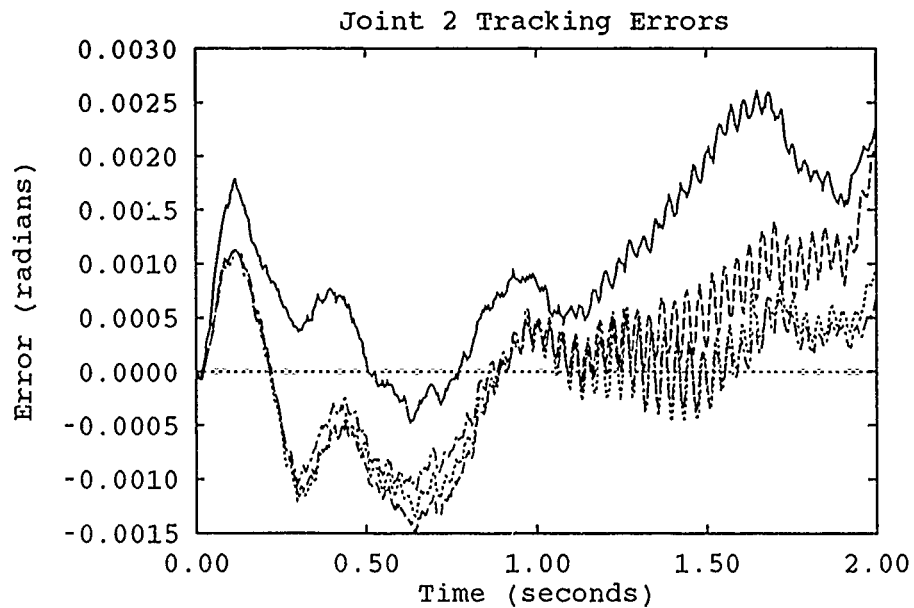


Figure H.14. 19-Parameter Initialized Adaptation Runs - Trajectory 5

—	First Run	Fourth Run
- - - -	Second Run	- · - · -	Fifth Run

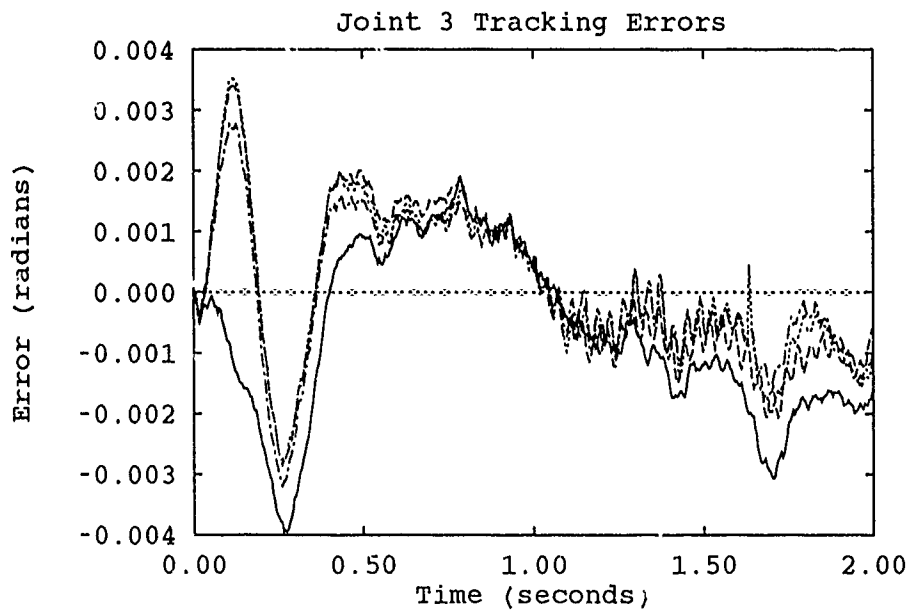


Figure H.15. 19-Parameter Initialized Adaptation Runs - Trajectory 5

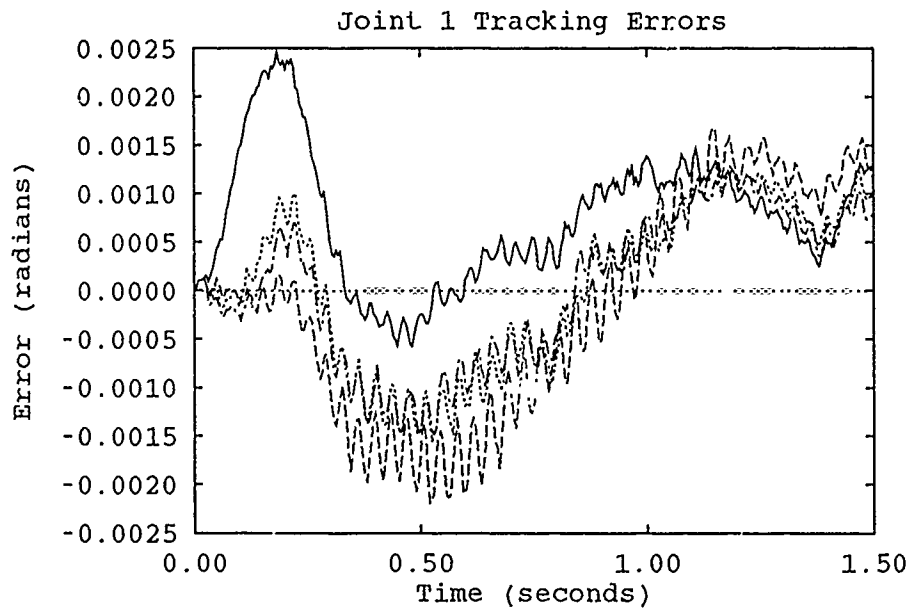


Figure H.16. 19-Parameter Initialized Adaptation Runs - Trajectory 1

—	First Run	Fourth Run
- - -	Second Run	- . - . -	Fifth Run

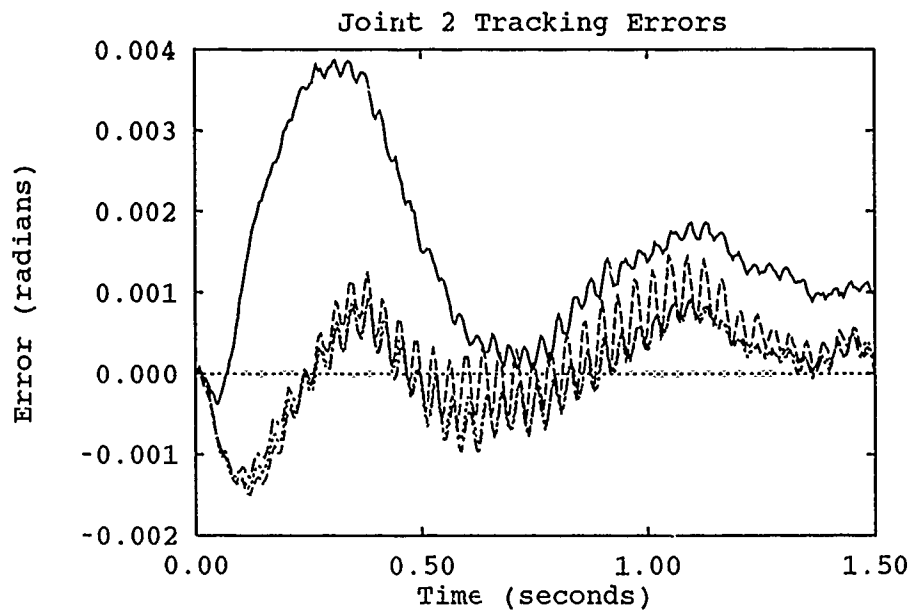


Figure H.17. 19-Parameter Initialized Adaptation Runs - Trajectory 1

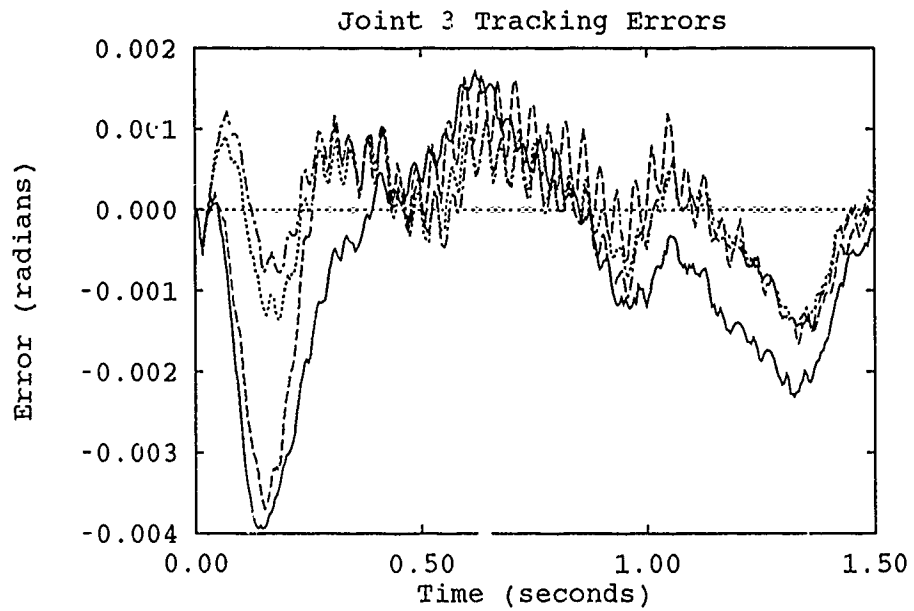


Figure H.18. 19-Parameter Initialized Adaptation Runs - Trajectory 1

—	First Run	Fourth Run
- - -	Second Run	- . - .	Fifth Run

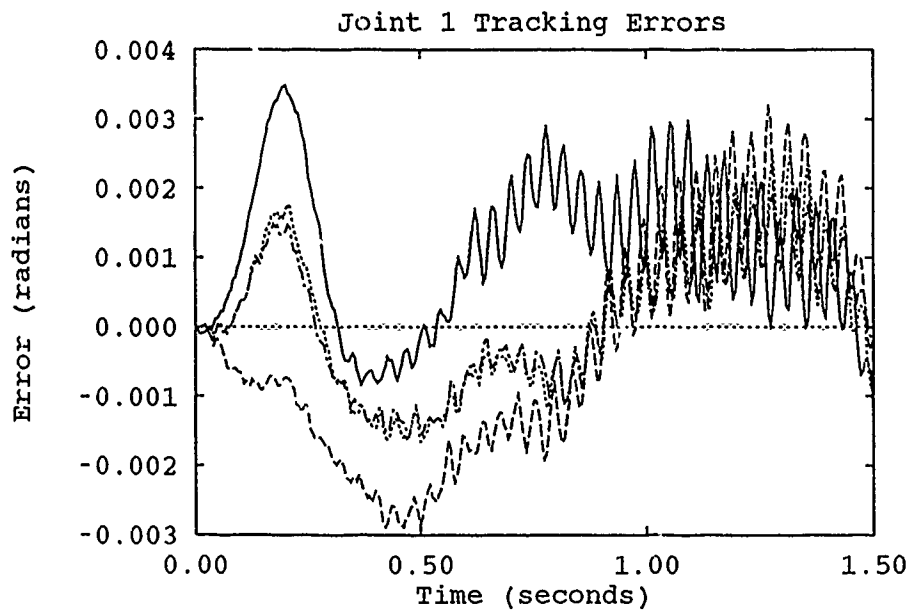


Figure H.19. 19-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

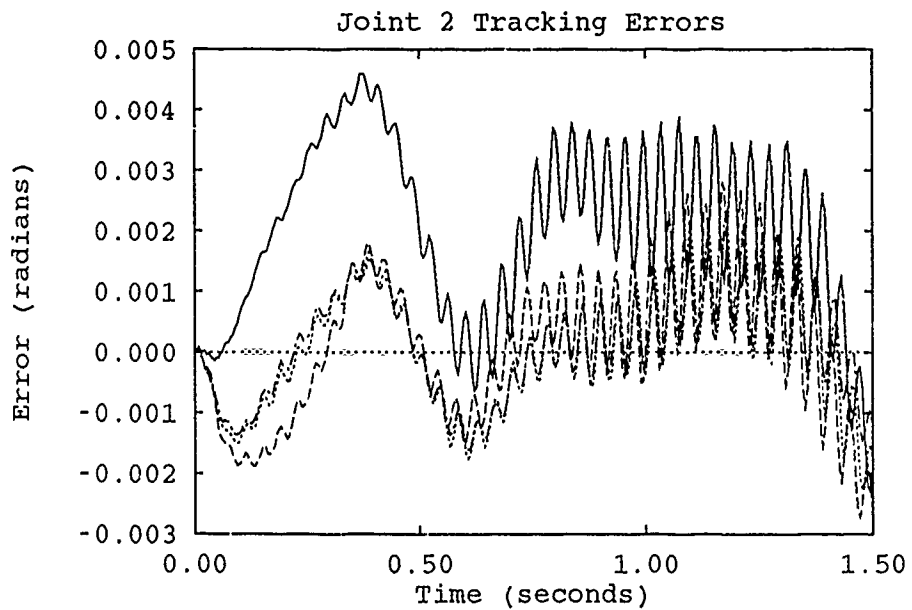


Figure H.20. 19-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

—	First Run	Fourth Run
- - - -	Second Run	- . - . -	Fifth Run

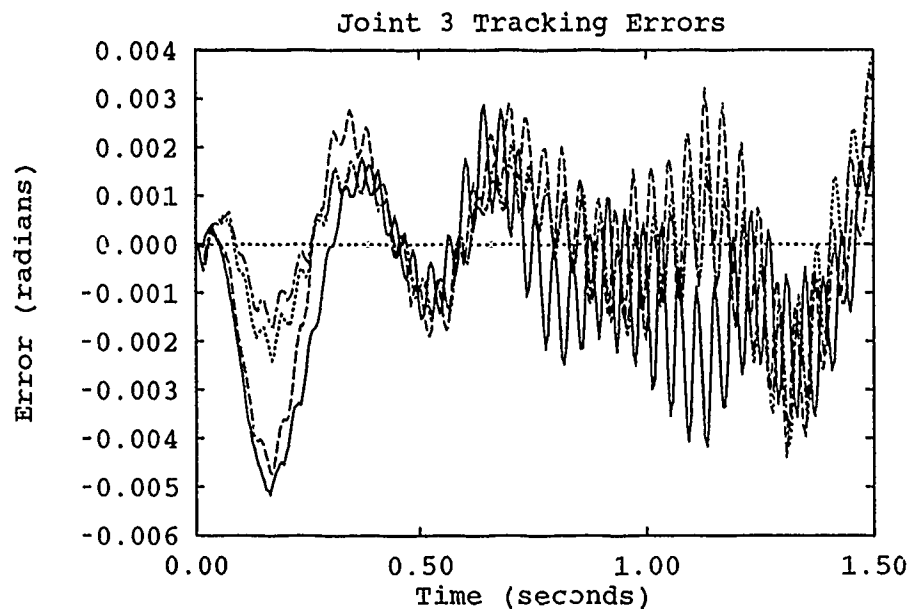


Figure H.21. 19-Parameter Initialized Adaptation Runs - Trajectory 1 w/ Payload

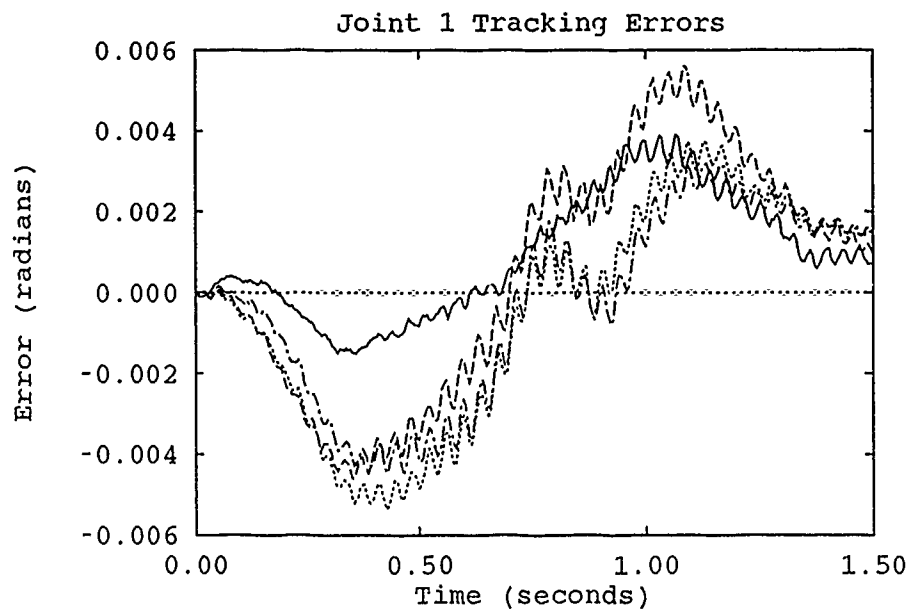


Figure H.22. 19-Parameter Initialized Adaptation Runs - Trajectory 6

—	First Run	Fourth Run
- - -	Second Run	- . - .	Fifth Run

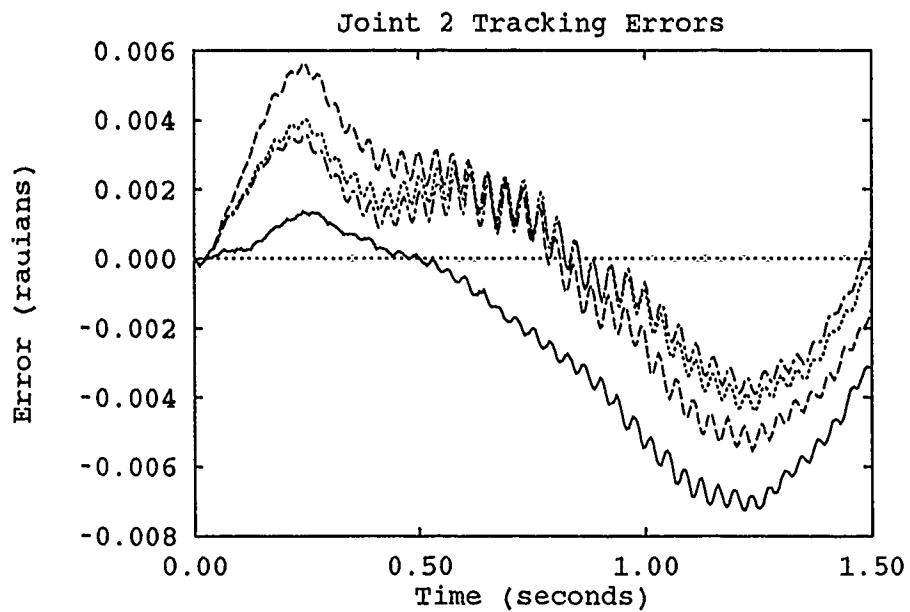


Figure H.23. 19-Parameter Initialized Adaptation Runs - Trajectory 6

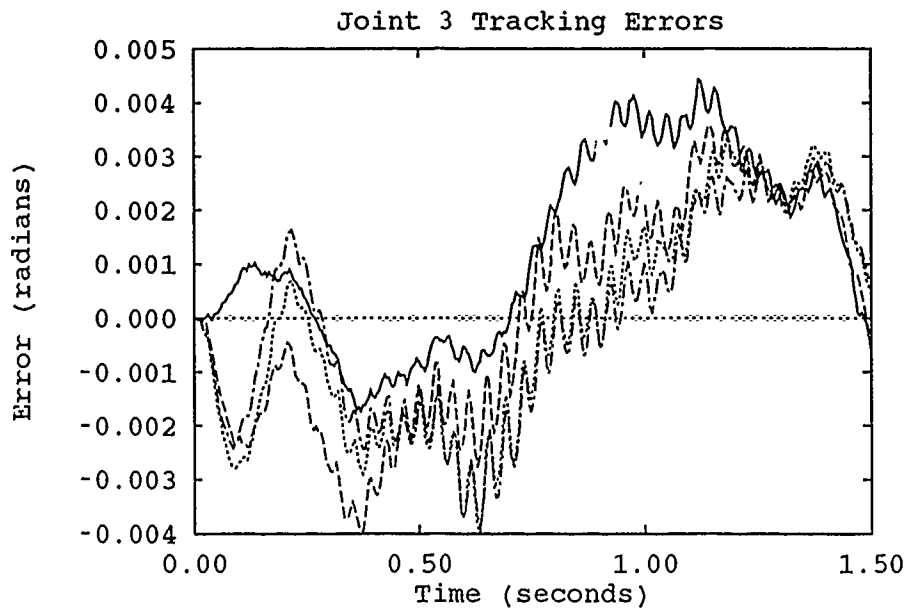


Figure H.24. 19-Parameter Initialized Adaptation Runs - Trajectory 6

—	First Run	Fourth Run
----	Second Run	- - - -	Fifth Run

Appendix I. *Comparisons of Single Initialized Runs*

This section contains a comparison of the first runs for 13-, 16-, and 19-parameter AMBC controllers for each test trajectory using initialized parameters.

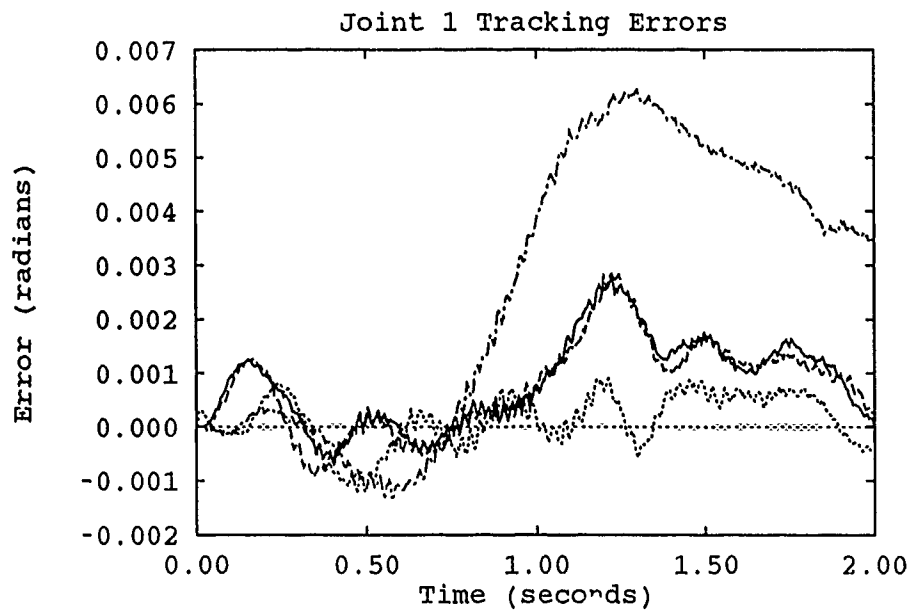


Figure I.1. Comparison of Single Initialized Runs - Trajectory 2

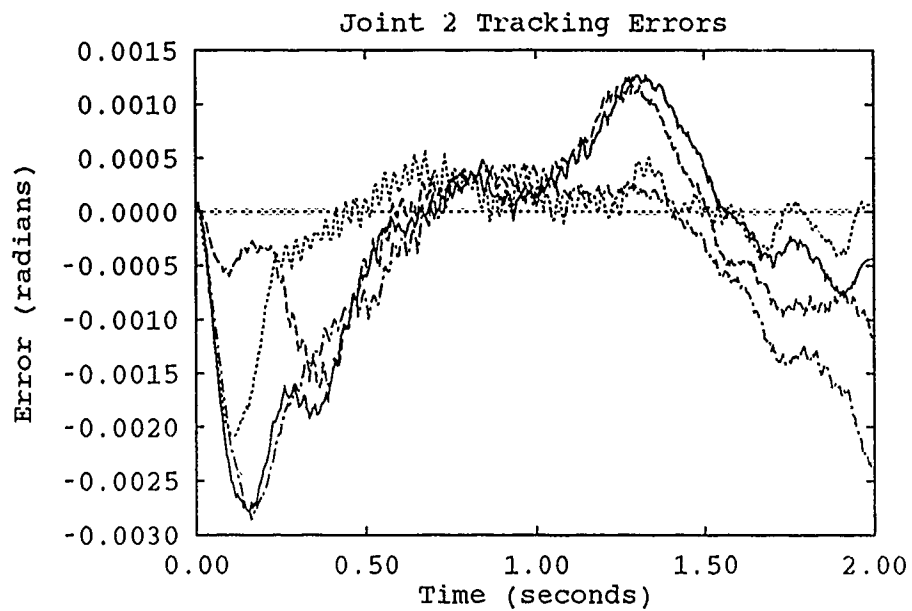


Figure I.2. Comparison of Single Initialized Runs - Trajectory 2

—	19 Parameters	13 Parameters
---	16 Parameters	- - - -	SMBC

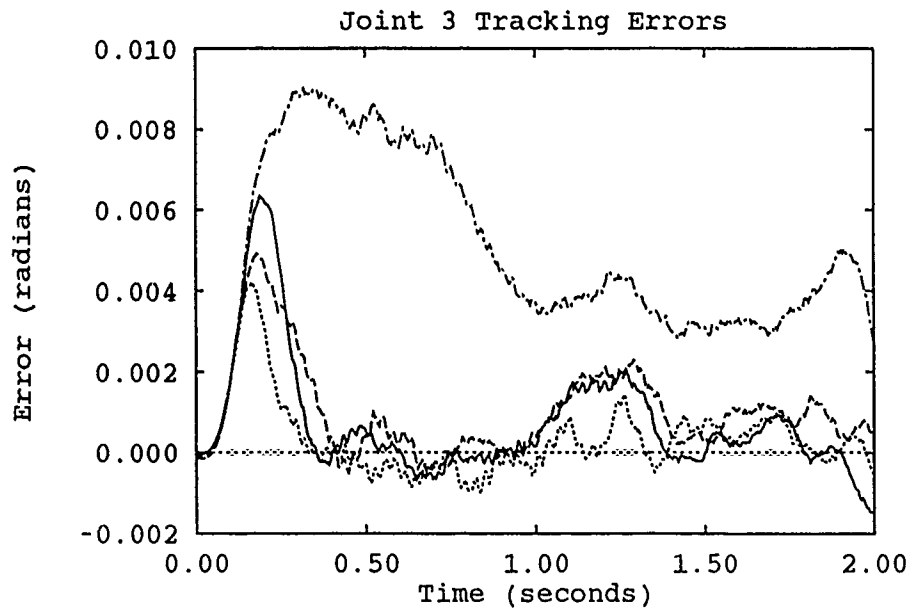


Figure I.3. Comparison of Single Initialized Runs - Trajectory 2

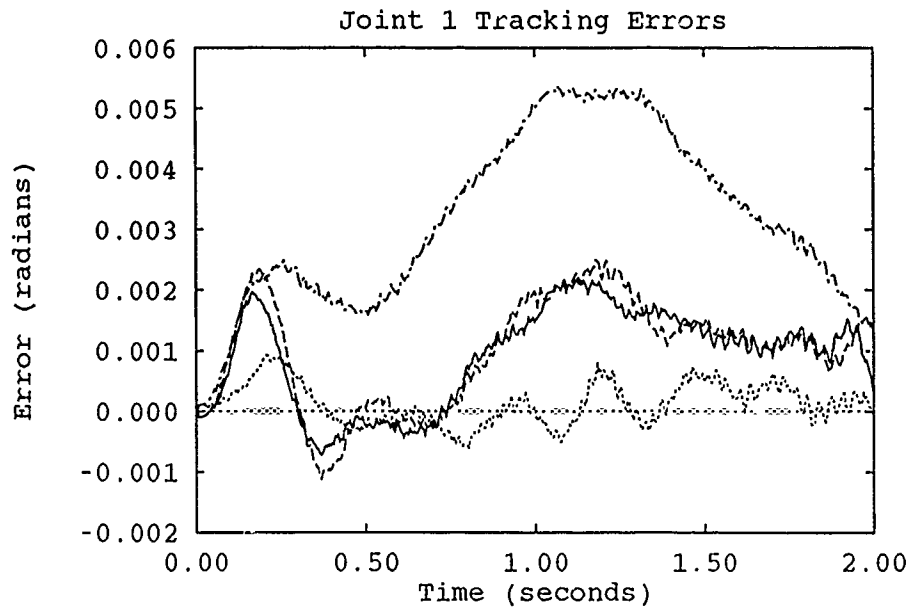


Figure I.4. Comparison of Single Initialized Runs - Trajectory 3

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

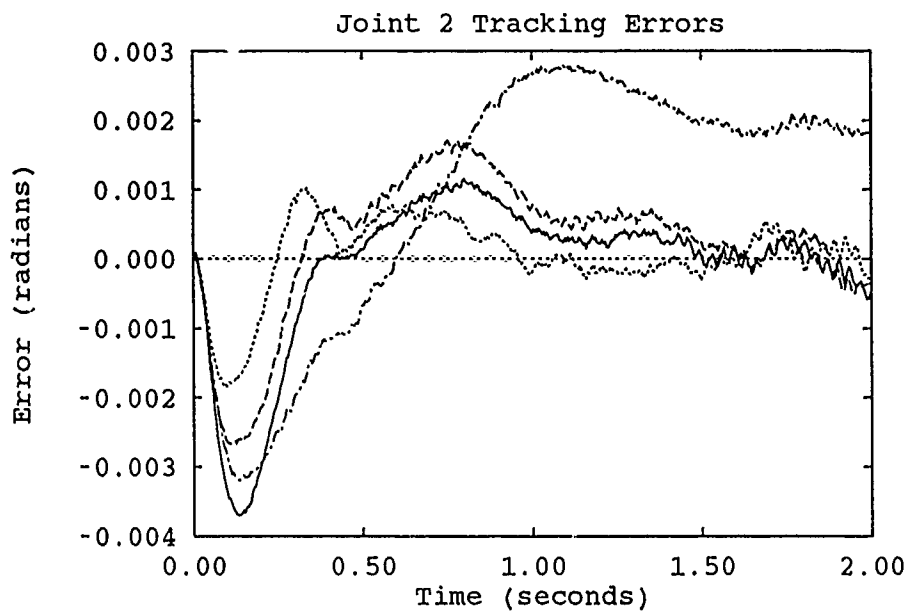


Figure I.5. Comparison of Single Initialized Runs - Trajectory 3

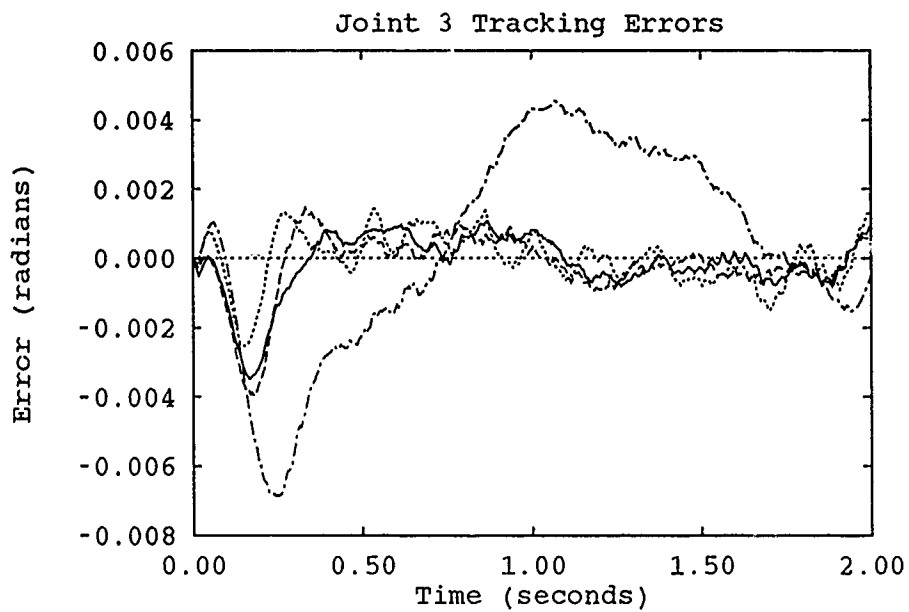


Figure I.6. Comparison of Single Initialized Runs - Trajectory 3

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

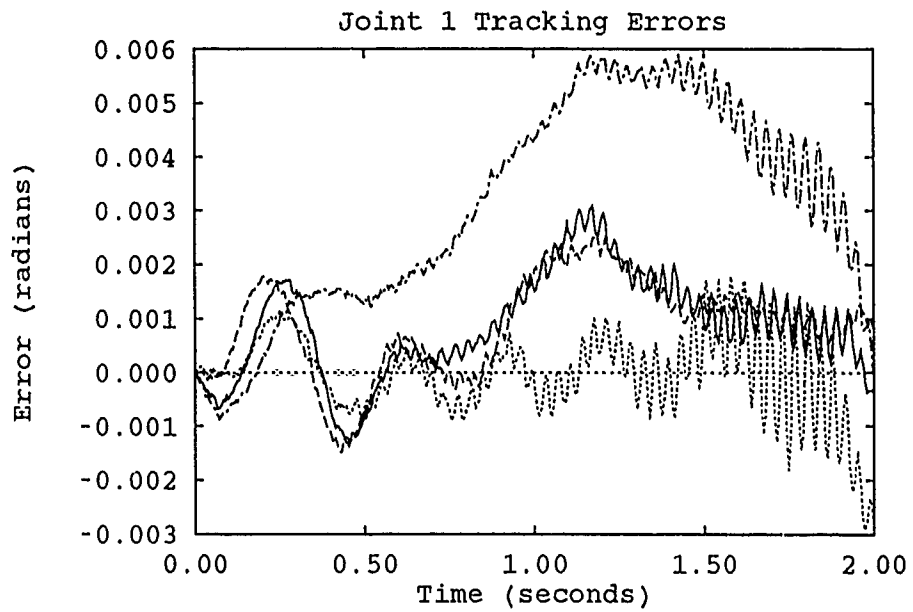


Figure I.7. Comparison of Single Initialized Runs - Trajectory 3 w/ Payload

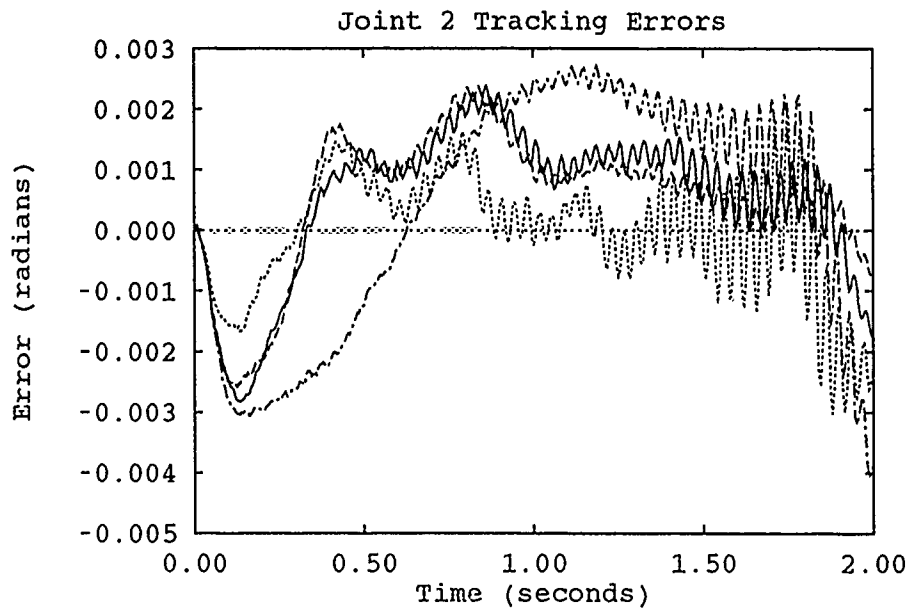


Figure I.8. Comparison of Single Initialized Runs - Trajectory 3 w/ Payload

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

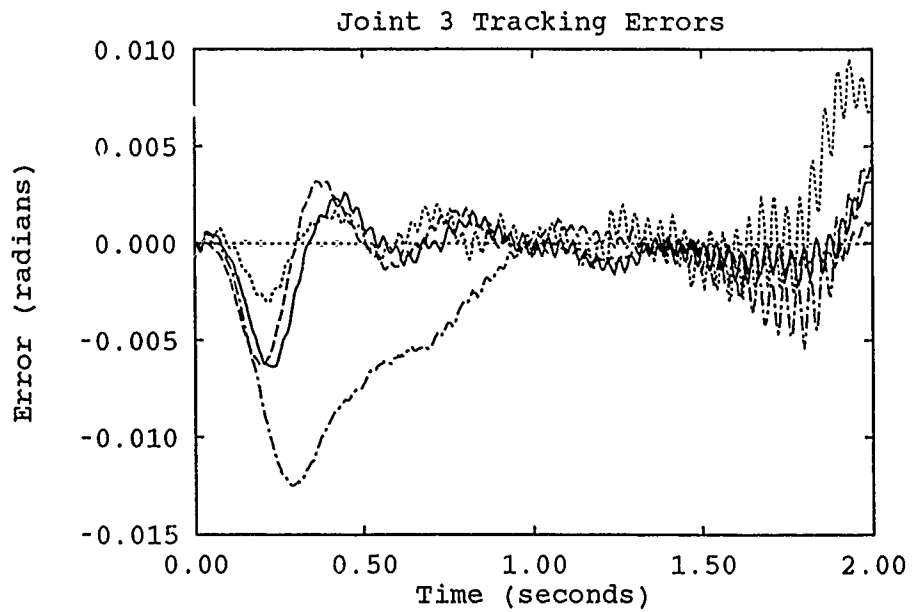


Figure I.9. Comparison of Single Initialized Runs - Trajectory 3 w/ Payload

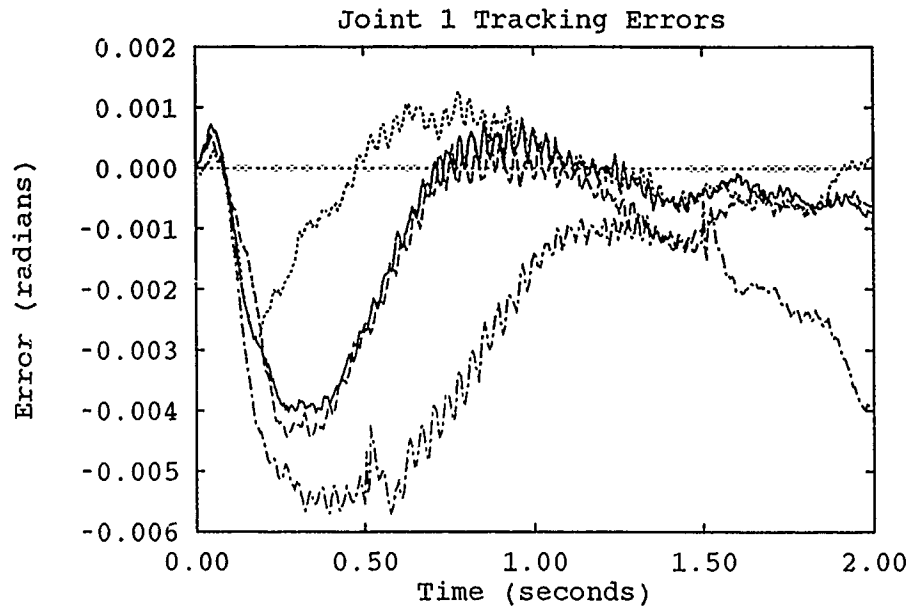


Figure I.10. Comparison of Single Initialized Runs - Trajectory 4

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

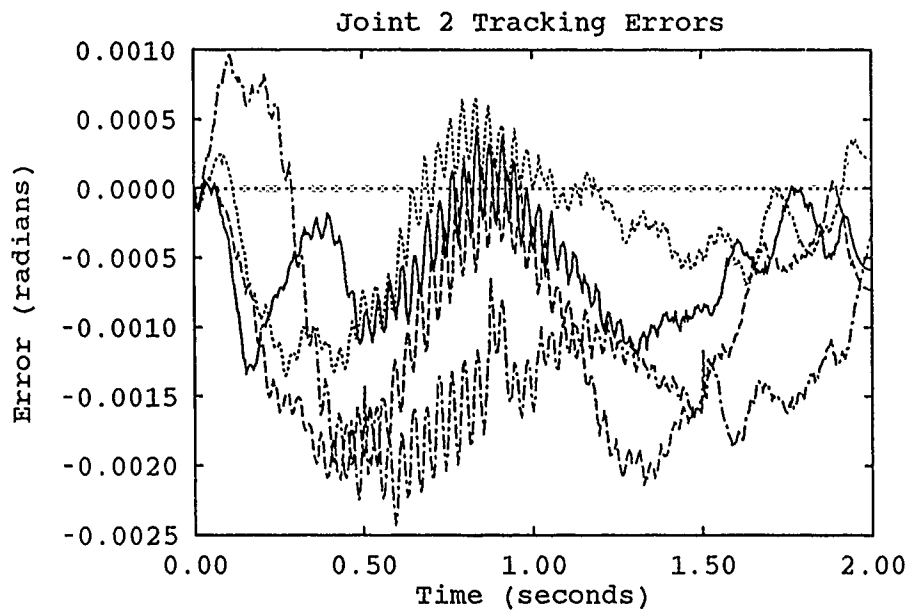


Figure I.11. Comparison of Single Initialized Runs - Trajectory 4

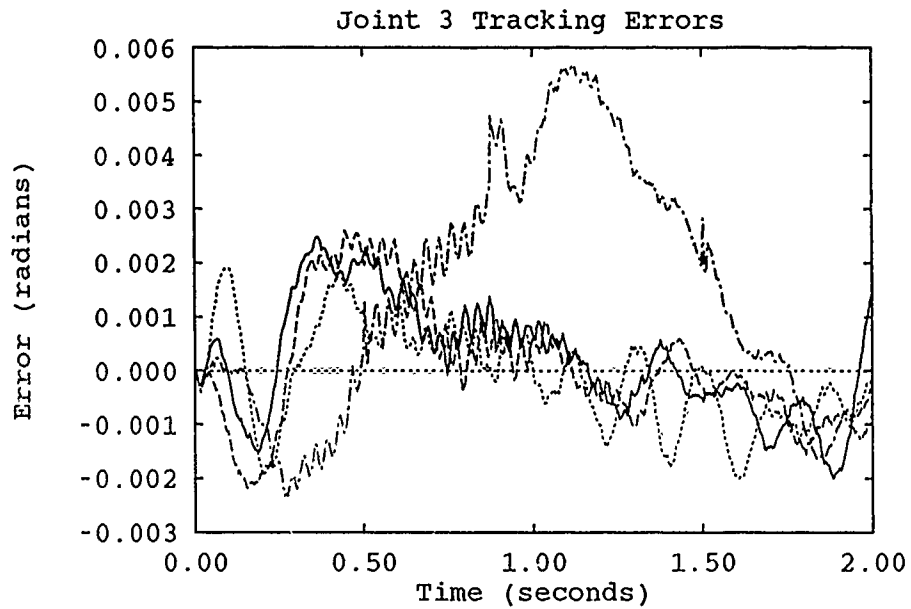


Figure I.12. Comparison of Single Initialized Runs - Trajectory 4

—	19 Parameters	13 Parameters
----	16 Parameters	-.-.-.	SMBC

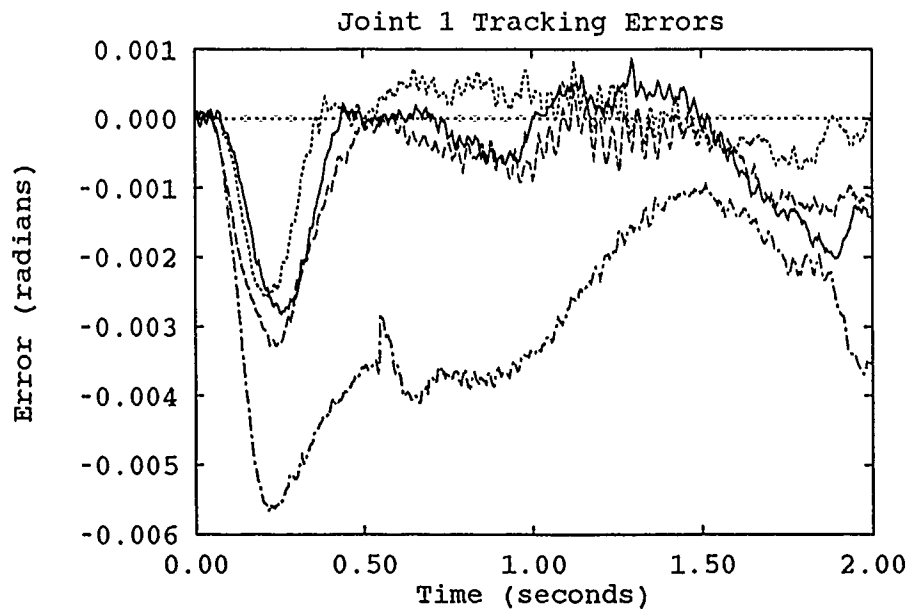


Figure I.13. Comparison of Single Initialized Runs - Trajectory 5

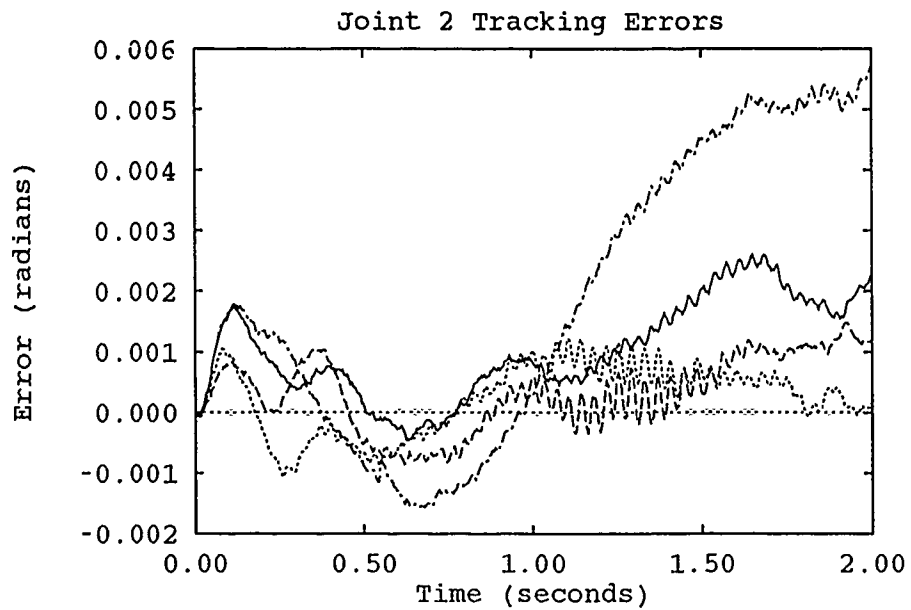


Figure I.14. Comparison of Single Initialized Runs - Trajectory 5

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - . -	SMBC

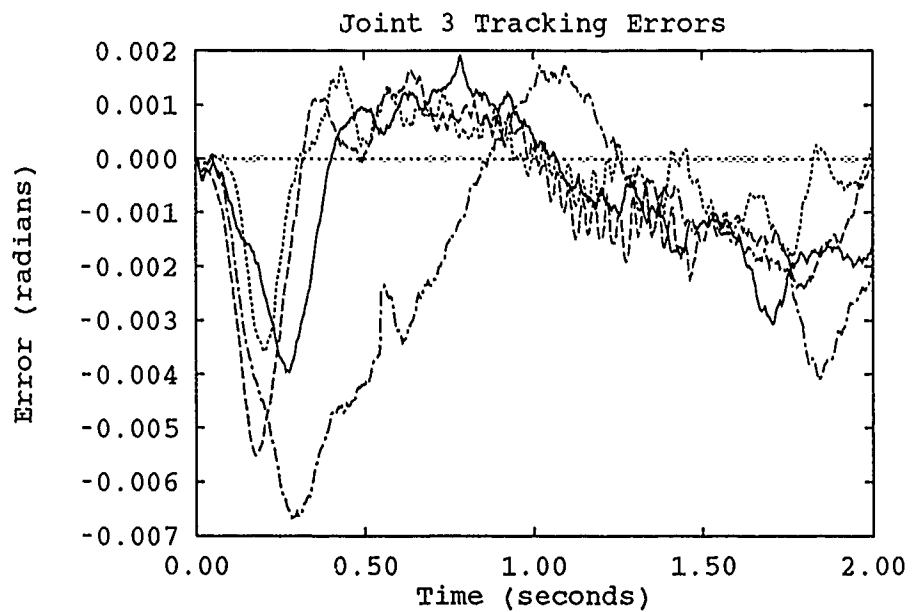


Figure I.15. Comparison of Single Initialized Runs - Trajectory 5

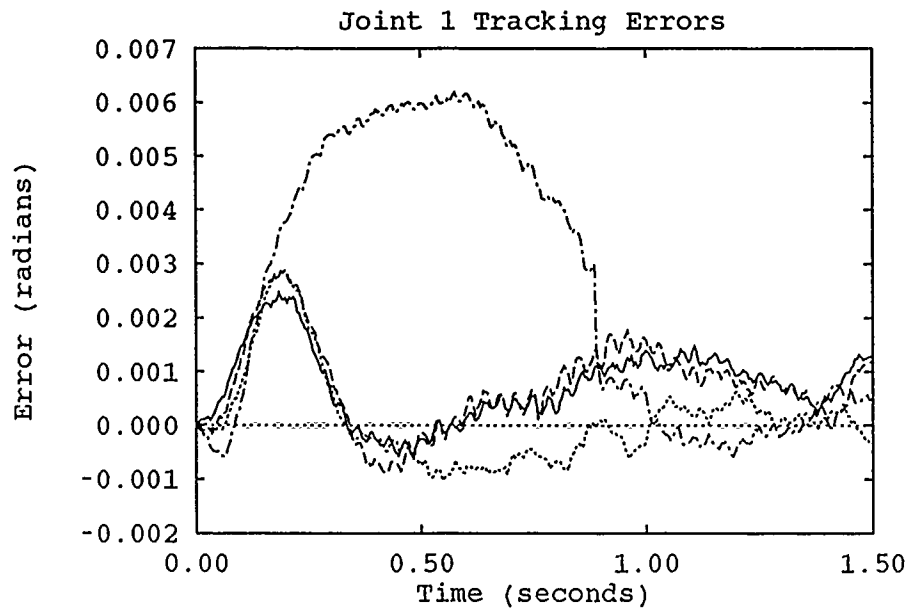


Figure I.16. Comparison of Single Initialized Runs - Trajectory 1

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

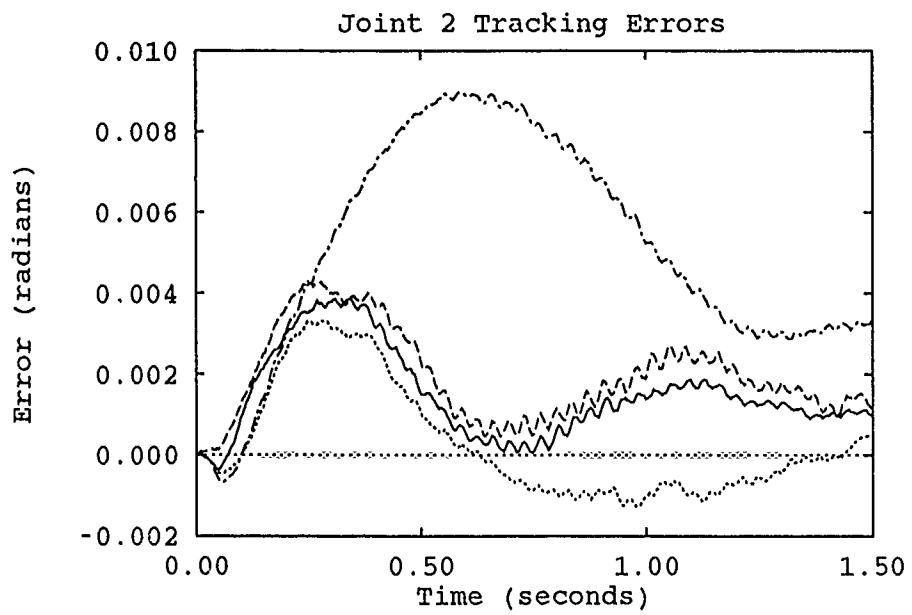


Figure I.17. Comparison of Single Initialized Runs - Trajectory 1

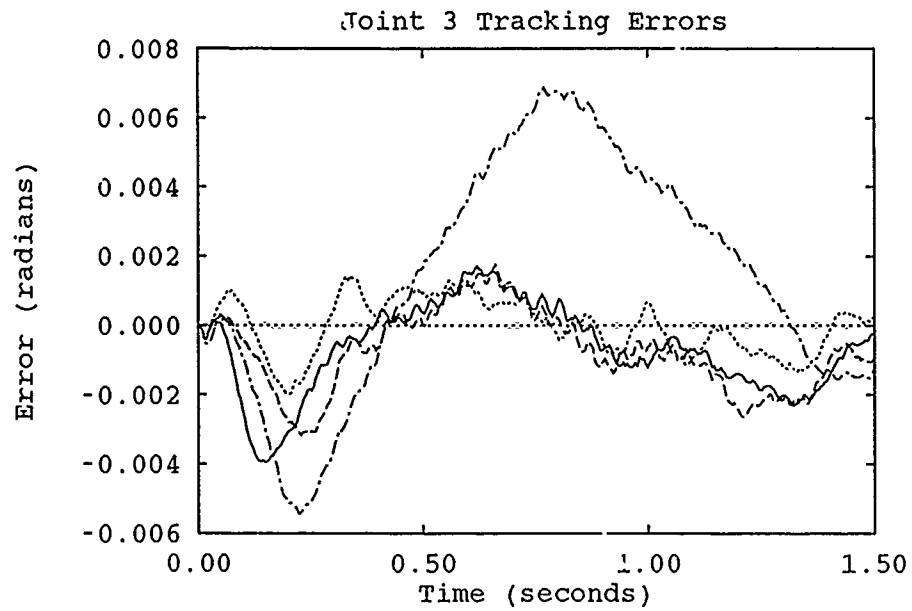


Figure I.18. Comparison of Single Initialized Runs - Trajectory 1

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - . -	SMBC

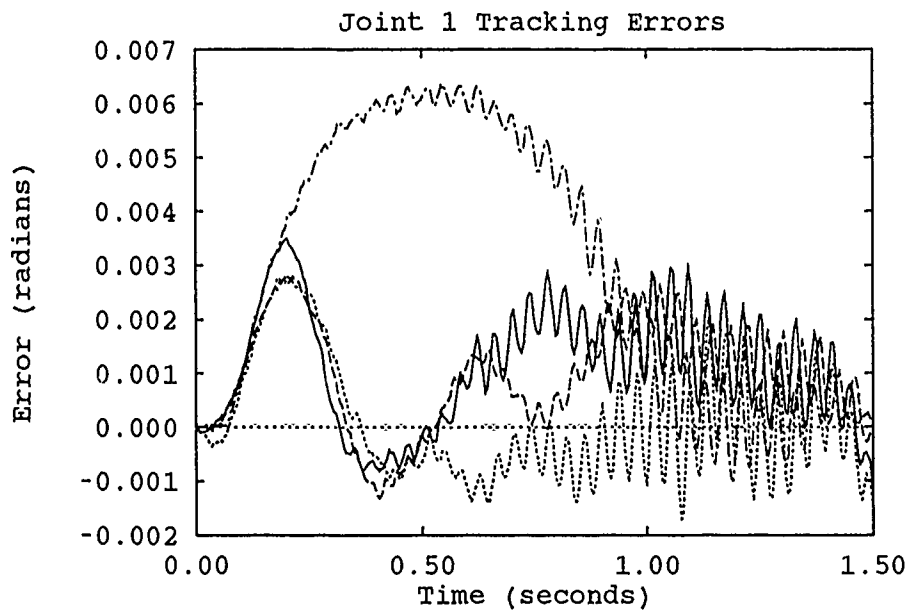


Figure I.19. Comparison of Single Initialized Runs - Trajectory 1 w/ Payload

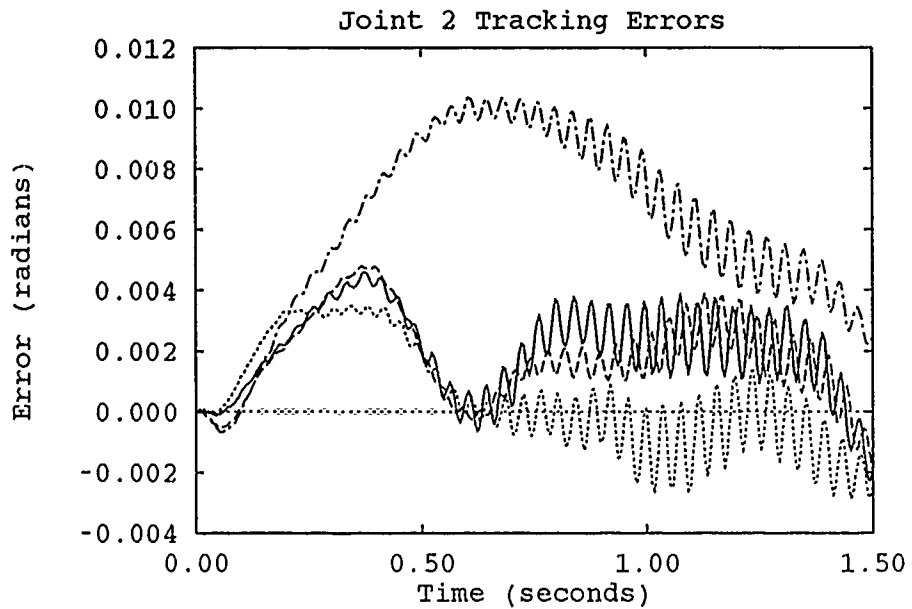


Figure I.20. Comparison of Single Initialized Runs - Trajectory 1 w/ Payload

—	19 Parameters	13 Parameters
- - - -	16 Parameters	- . - . -	SMBC

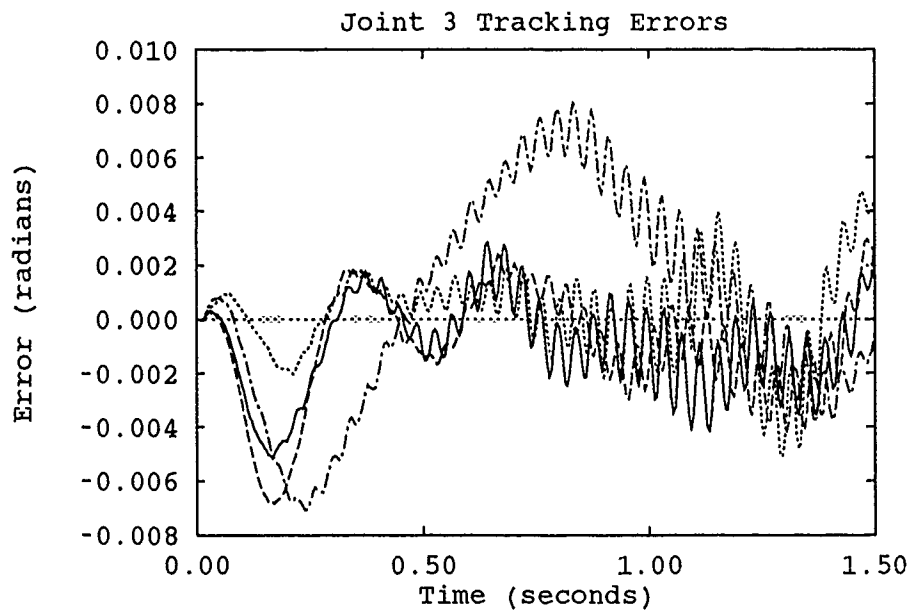


Figure I.21. Comparison of Single Initialized Runs - Trajectory 1 w/ Payload

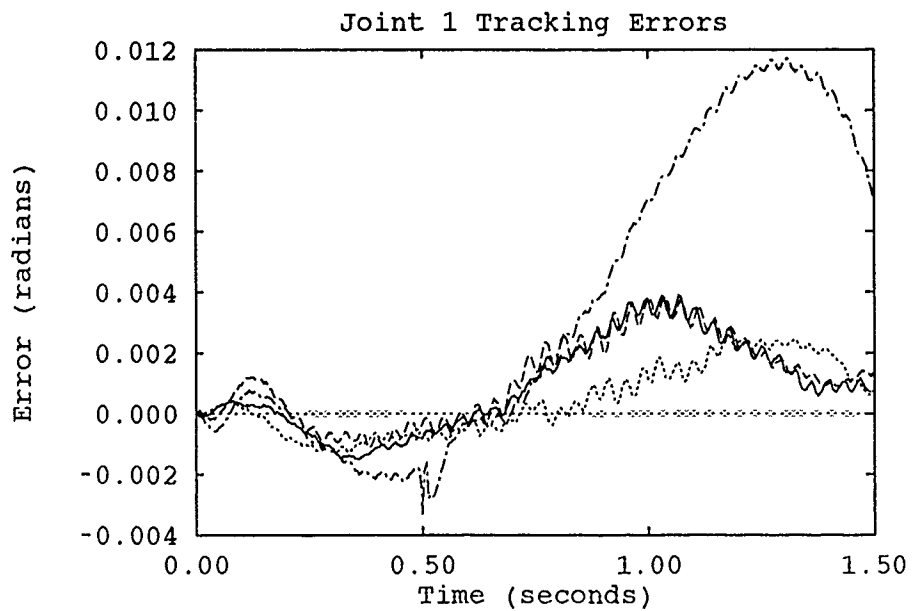


Figure I.22. Comparison of Single Initialized Runs - Trajectory 6

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - . -	SMBC

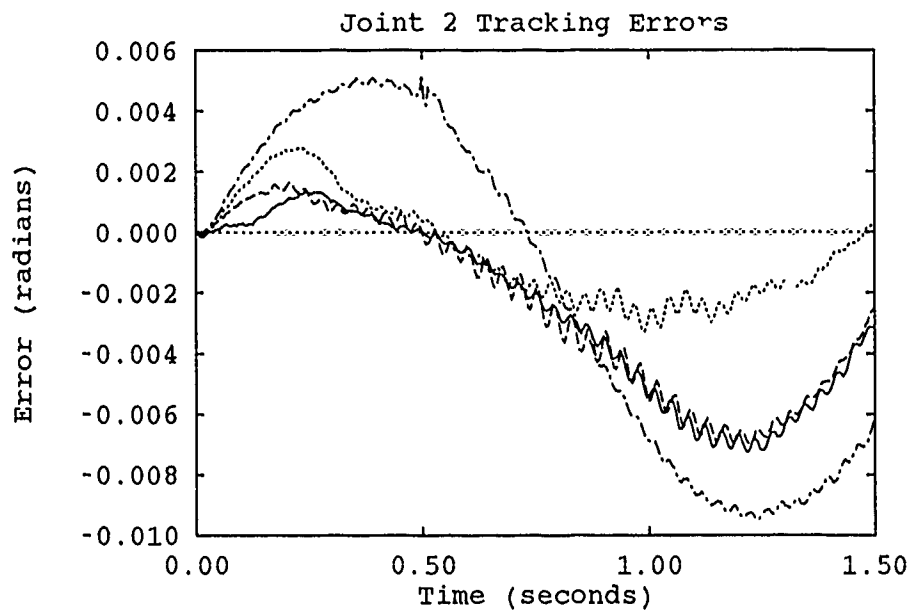


Figure I.23. Comparison of Single Initialized Runs - Trajectory 6

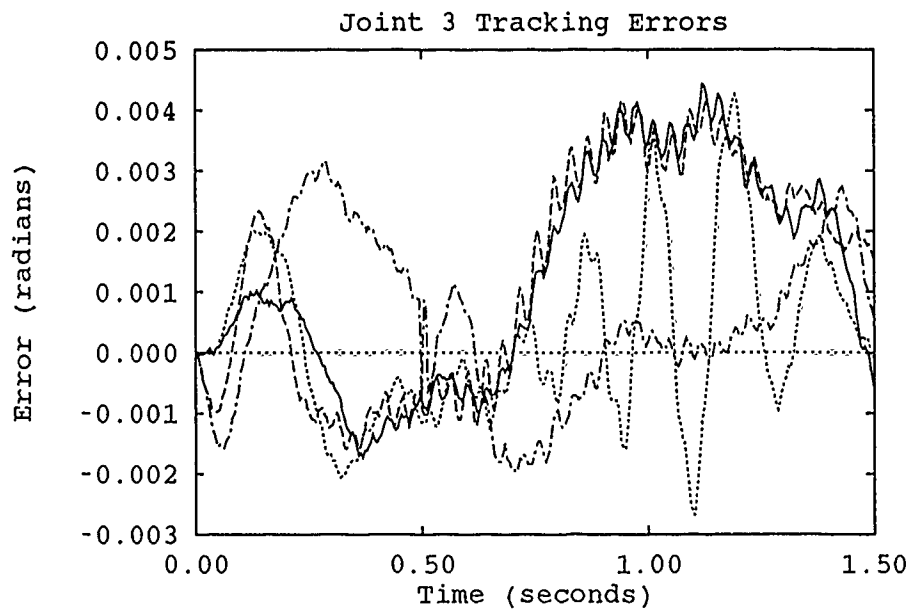


Figure I.24. Comparison of Single Initialized Runs - Trajectory 6

—	19 Parameters	13 Parameters
- - -	16 Parameters	- . - .	SMBC

Appendix J. *Comparisons of Initialized Runs After Learning*

This section contains a comparison of the 13-, 16-, and 19-parameter adaptation runs for each test trajectory using initialized parameters. Each trajectory was run five times to allow the controller to adapt.

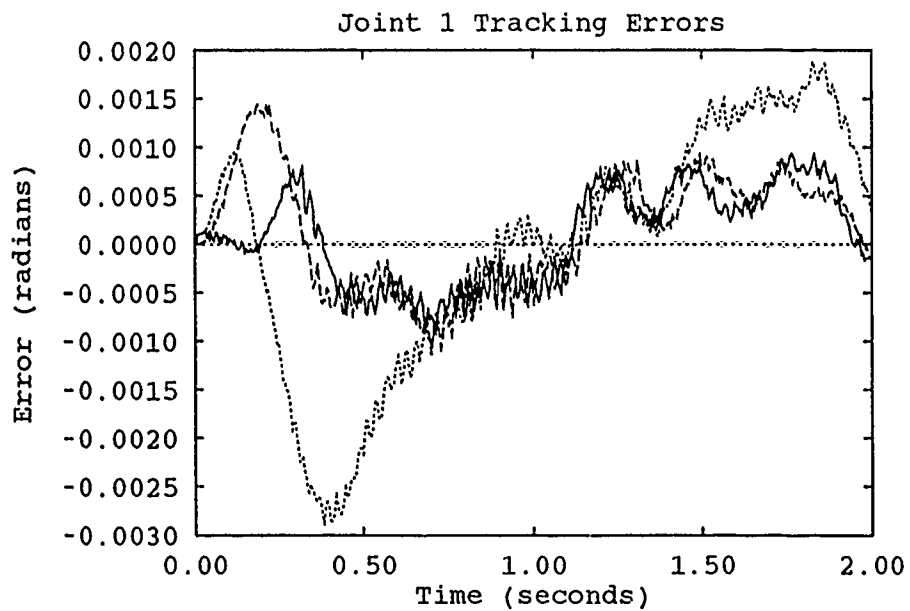


Figure J.1. Comparison of Initialized Runs After Learning - Trajectory 2

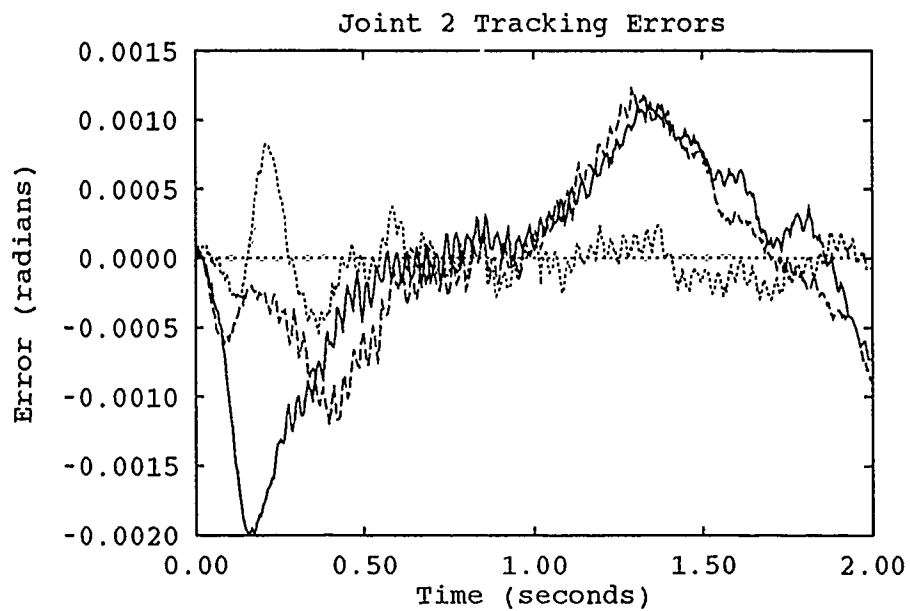


Figure J.2. Comparison of Initialized Runs After Learning - Trajectory 2

—	19 Parameters	13 Parameters
- - -	16 Parameters		

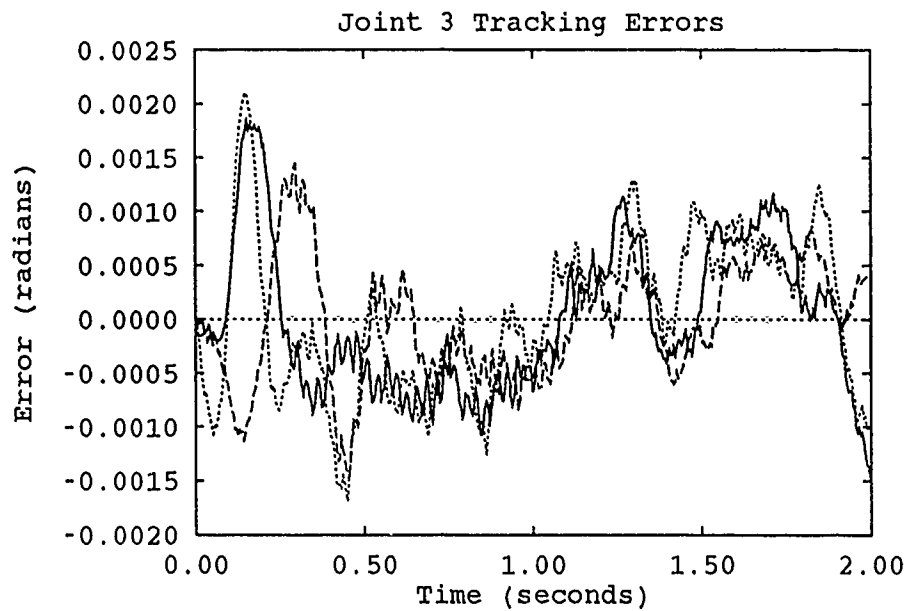


Figure J.3. Comparison of Initialized Runs After Learning - Trajectory 2

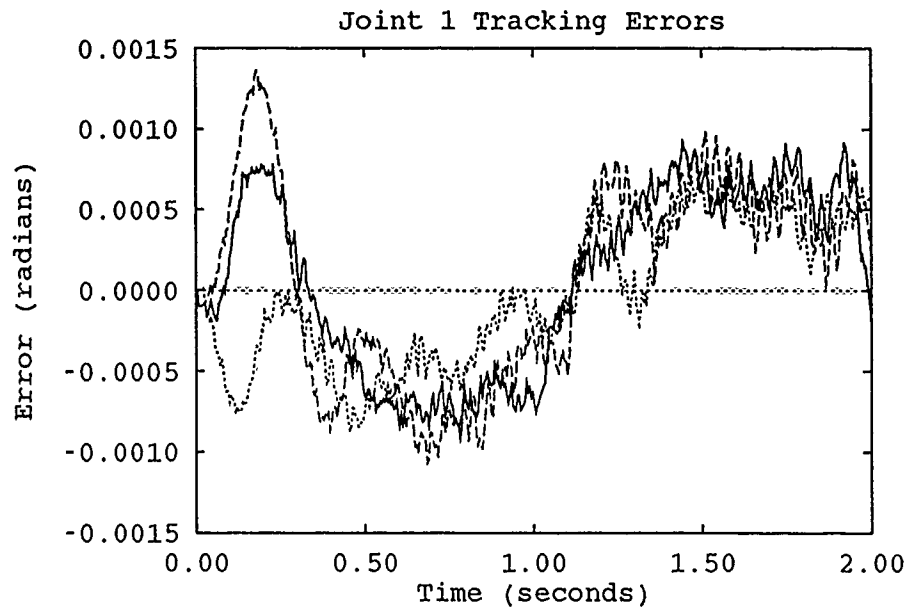


Figure J.4. Comparison of Initialized Runs After Learning - Trajectory 3

—	19 Parameters	13 Parameters
- - -	16 Parameters		

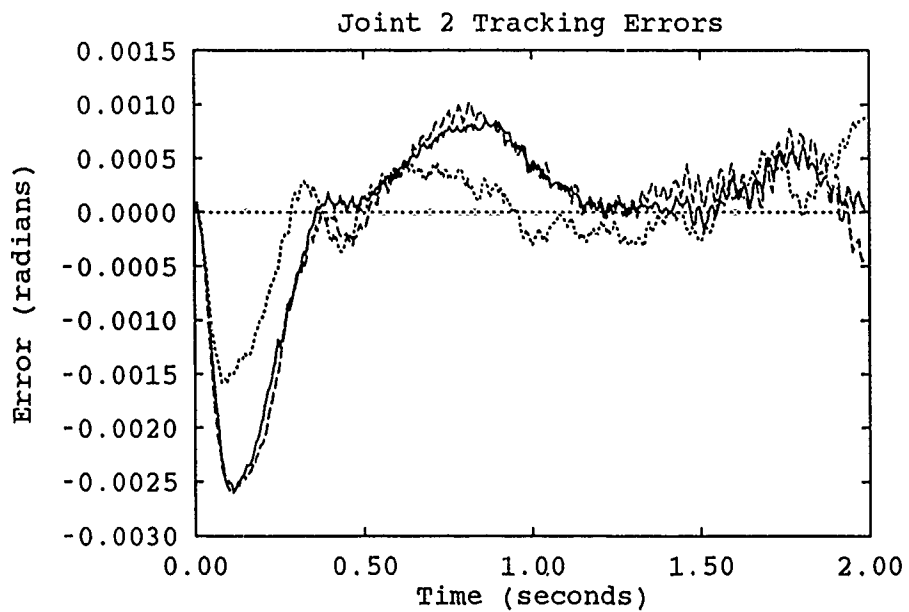


Figure J.5. Comparison of Initialized Runs After Learning - Trajectory 3

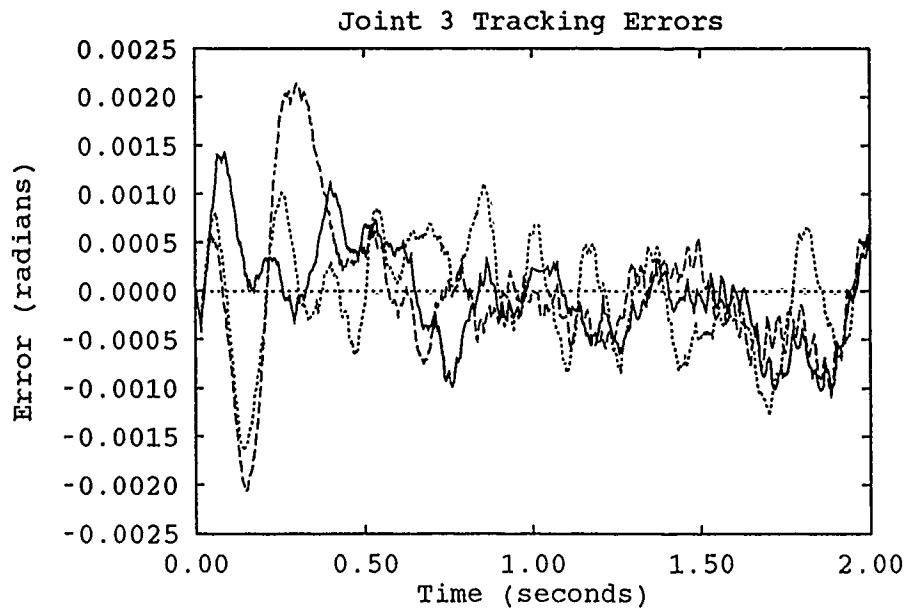


Figure J.6. Comparison of Initialized Runs After Learning - Trajectory 3

—	19 Parameters	13 Parameters
- - -	16 Parameters		

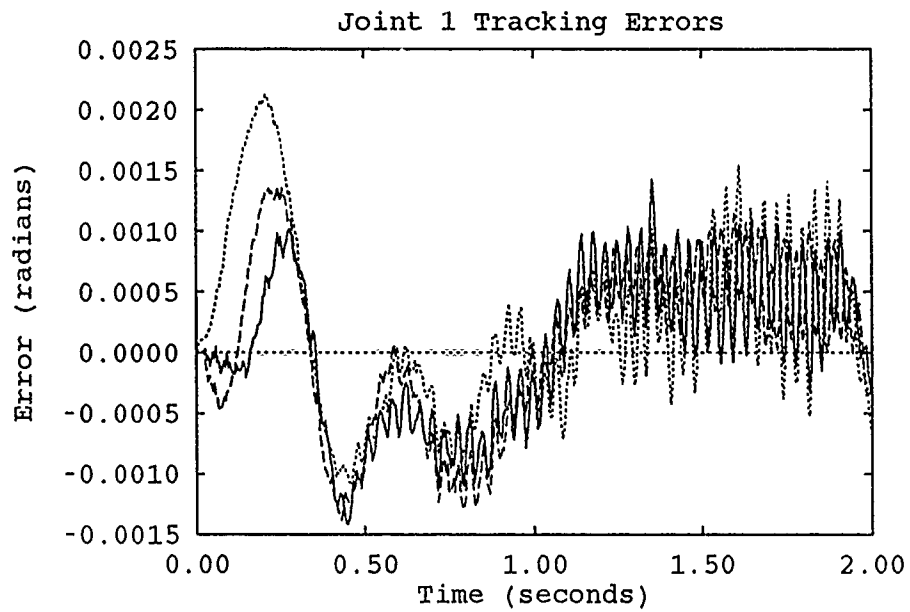


Figure J.7. Comparison of Initialized Runs After Learning - Trajectory 3 w/ Payload

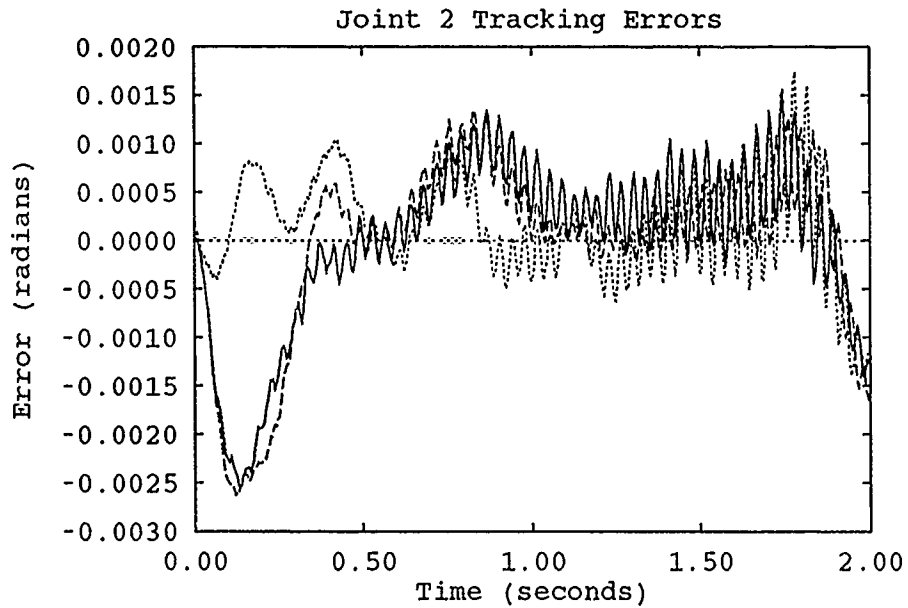


Figure J.8. Comparison of Initialized Runs After Learning - Trajectory 3 w/ Payload

——	19 Parameters	13 Parameters
----	16 Parameters		

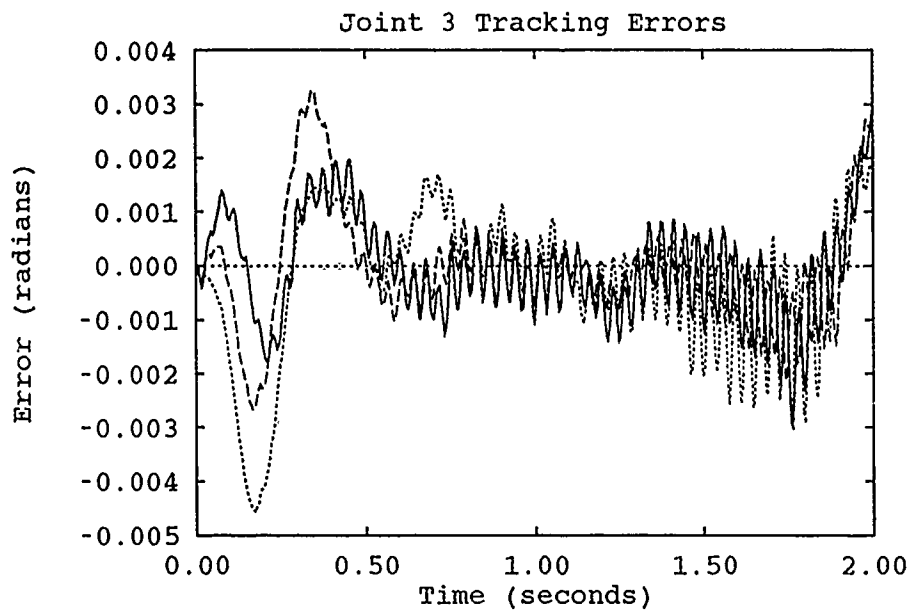


Figure J.9. Comparison of Initialized Runs After Learning - Trajectory 3 w/ Payload

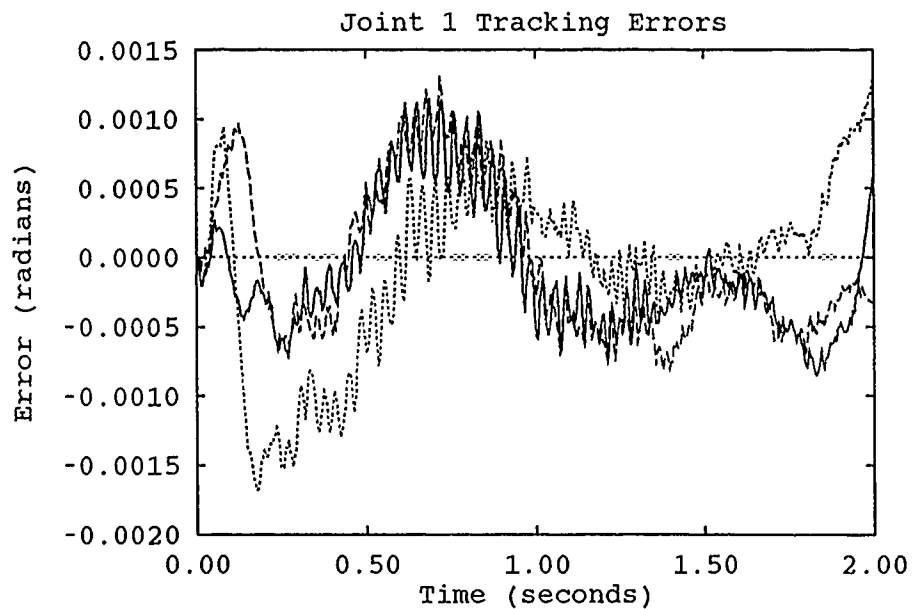


Figure J.10. Comparison of Initialized Runs After Learning - Trajectory 4

—	19 Parameters	13 Parameters
----	16 Parameters		

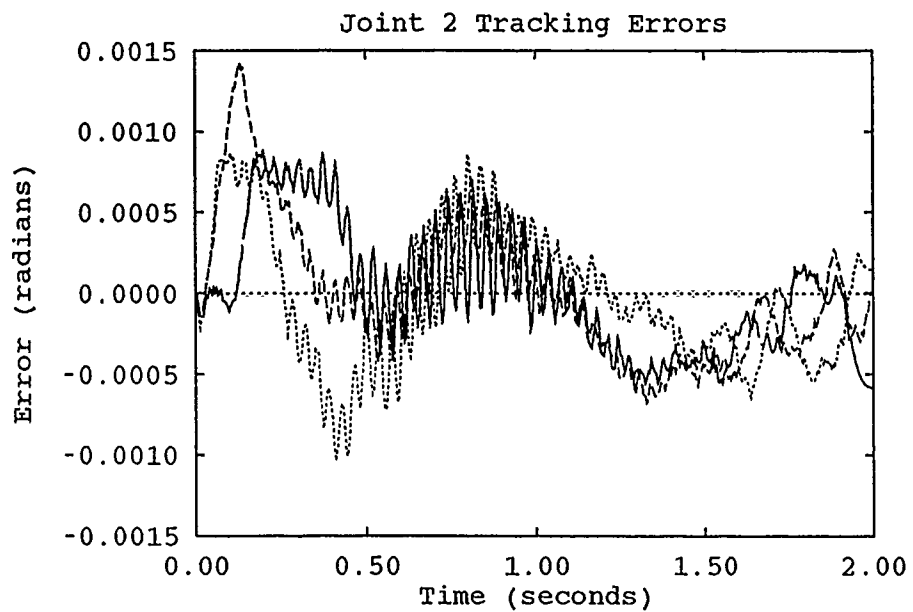


Figure J.11. Comparison of Initialized Runs After Learning - Trajectory 4

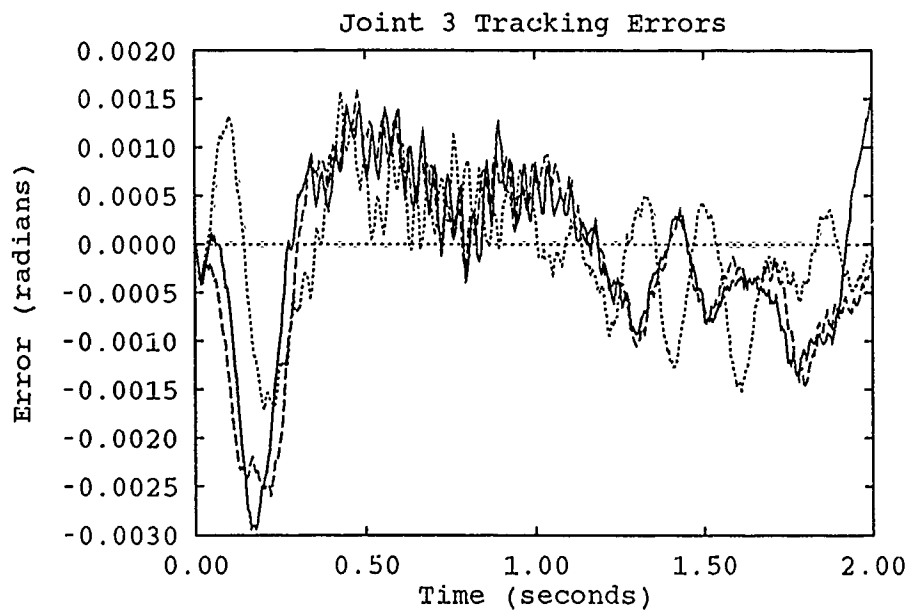


Figure J.12. Comparison of Initialized Runs After Learning - Trajectory 4

—	19 Parameters	13 Parameters
---	16 Parameters		

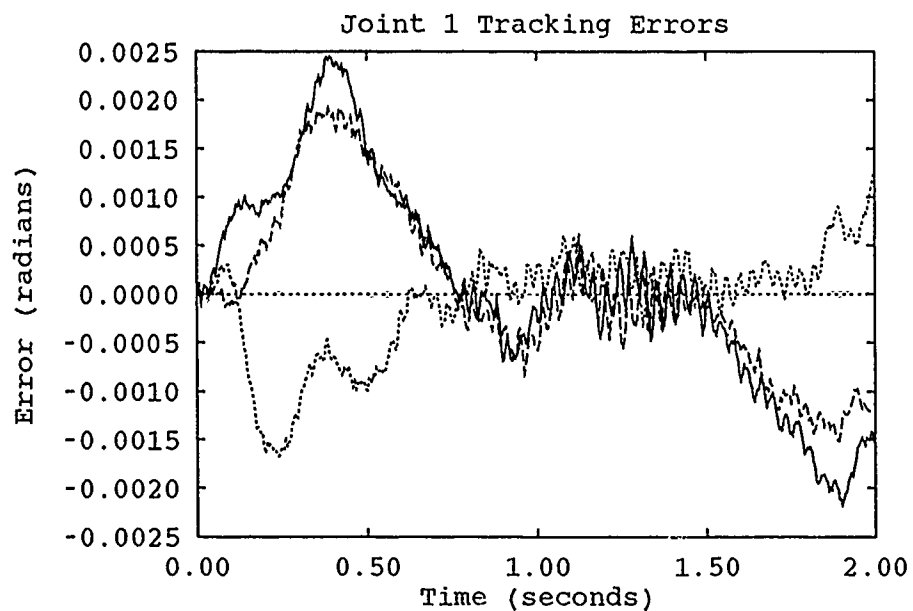


Figure J.13. Comparison of Initialized Runs After Learning - Trajectory 5

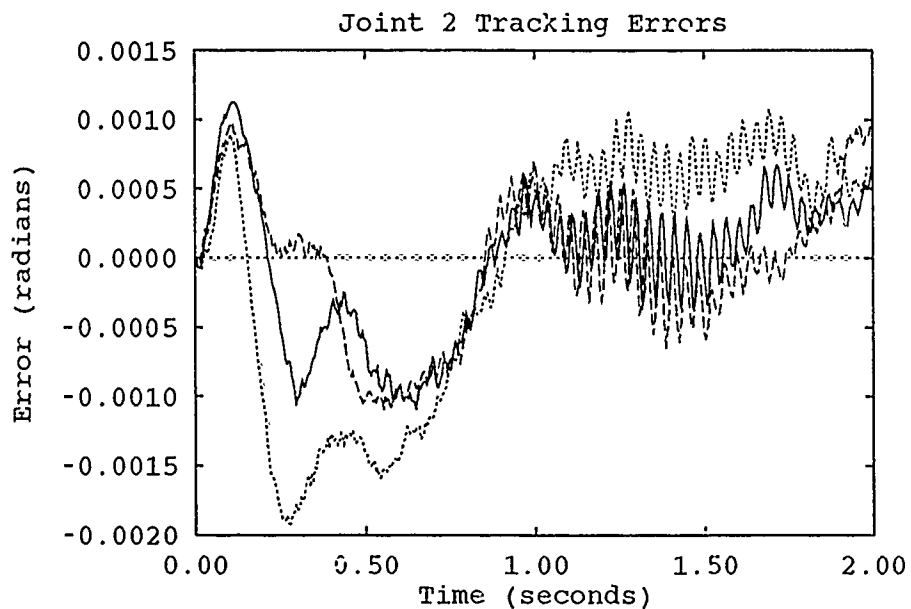


Figure J.14. Comparison of Initialized Runs After Learning - Trajectory 5

—	19 Parameters	13 Parameters
- - -	16 Parameters		

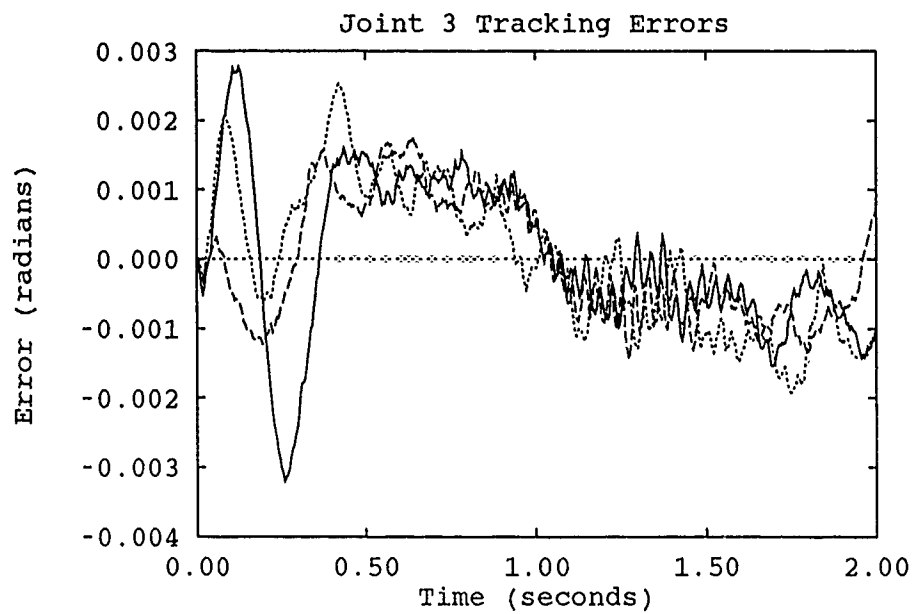


Figure J.15. Comparison of Initialized Runs After Learning - Trajectory 5

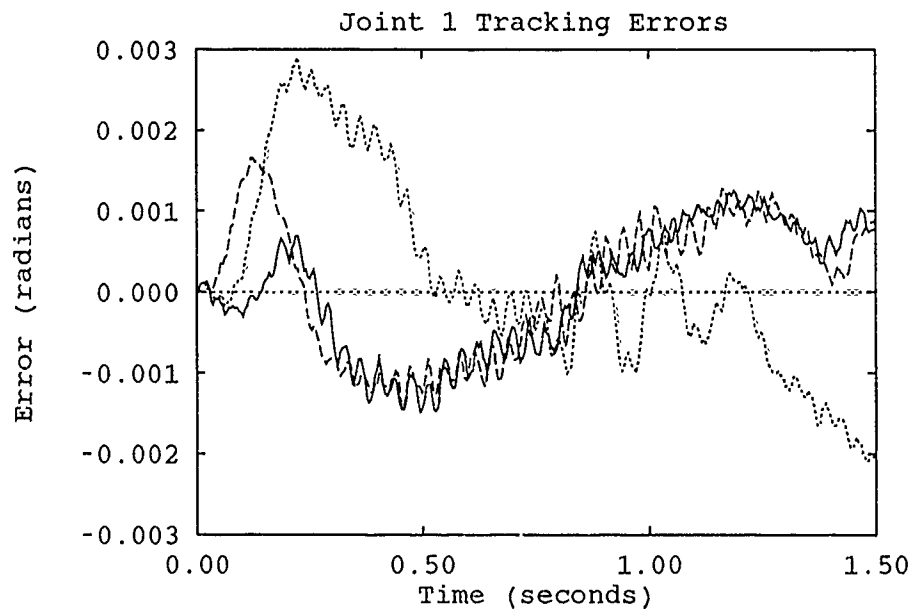


Figure J.16. Comparison of Initialized Runs After Learning - Trajectory 1

—	19 Parameters	13 Parameters
- - -	16 Parameters		

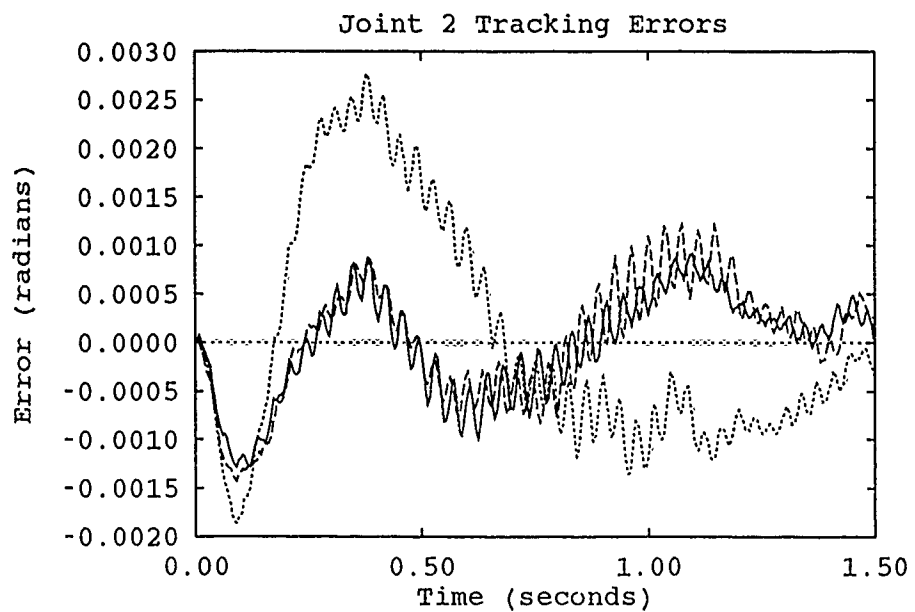


Figure J.17. Comparison of Initialized Runs After Learning - Trajectory 1

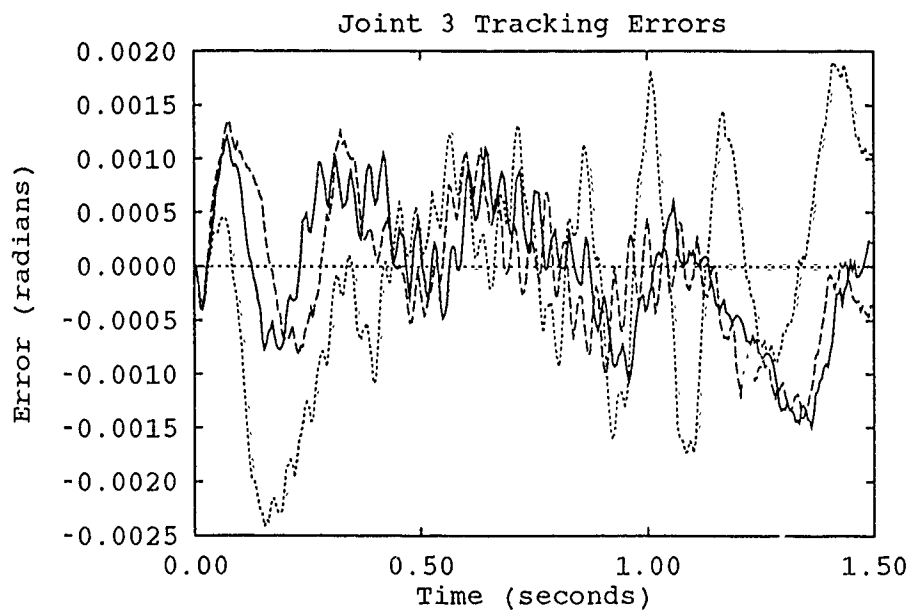


Figure J.18. Comparison of Initialized Runs After Learning - Trajectory 1

—	19 Parameters	13 Parameters
- - -	16 Parameters		

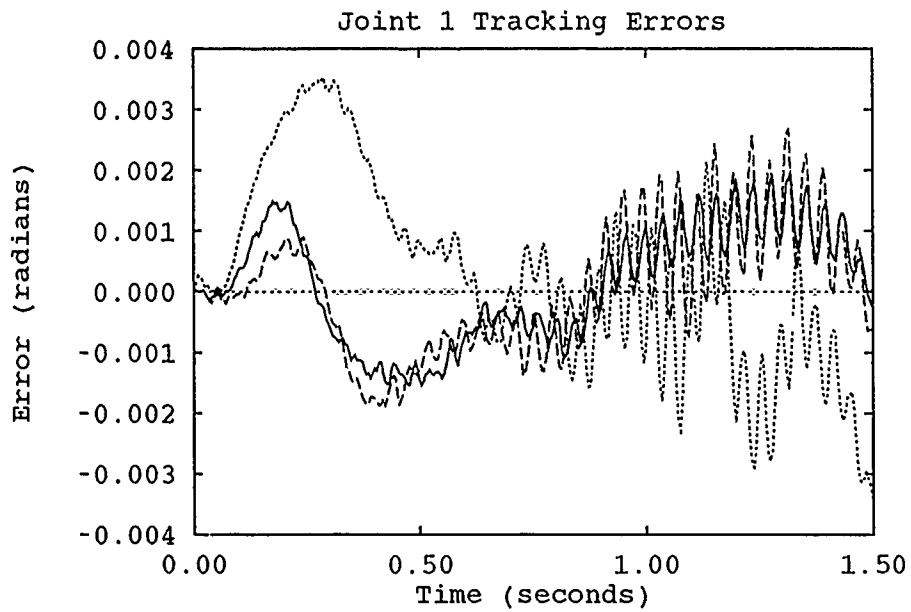


Figure J.19. Comparison of Initialized Runs After Learning - Trajectory 1 w/ Payload

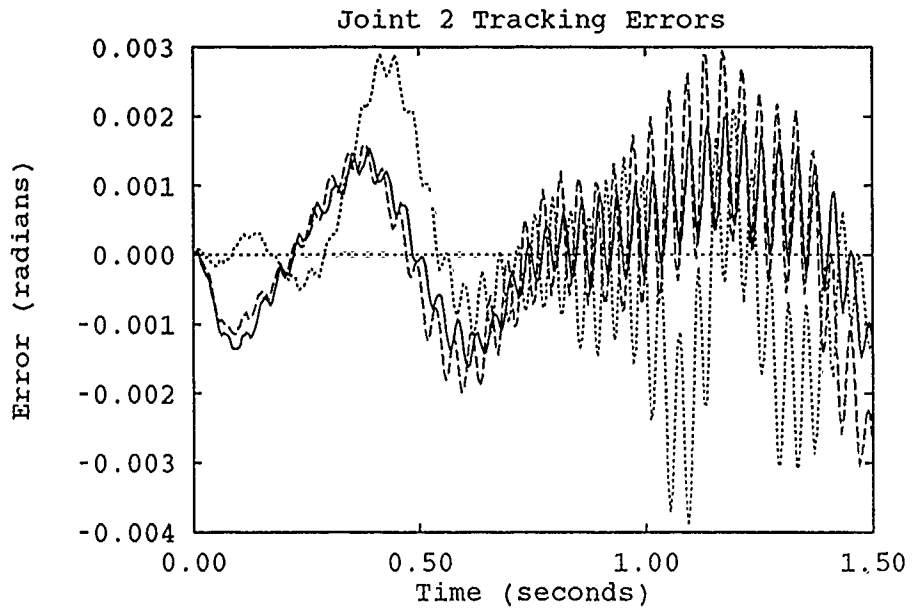


Figure J.20. Comparison of Initialized Runs After Learning - Trajectory 1 w/ Payload

—	19 Parameters	13 Parameters
- - - -	16 Parameters		

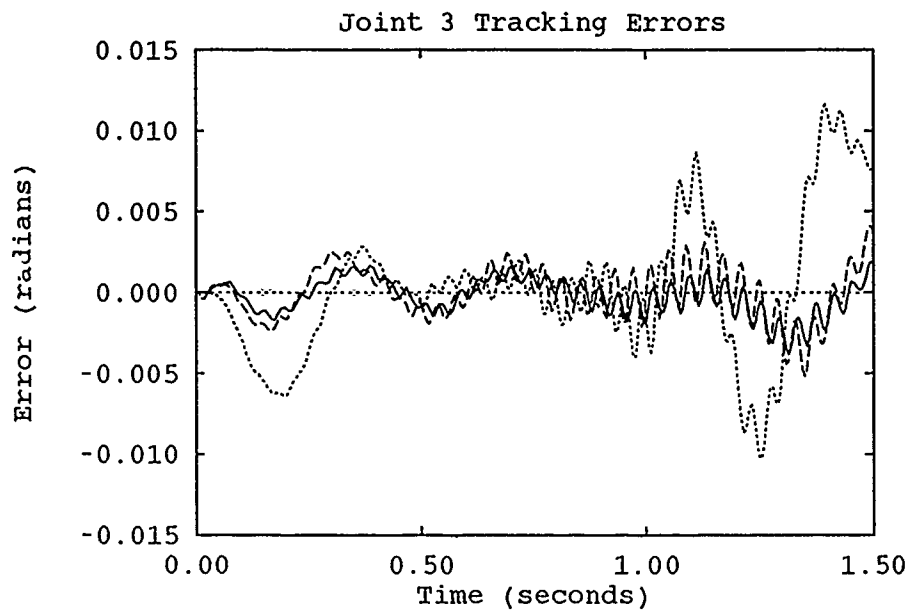


Figure J.21. Comparison of Initialized Runs After Learning - Trajectory 1 w/ Payload

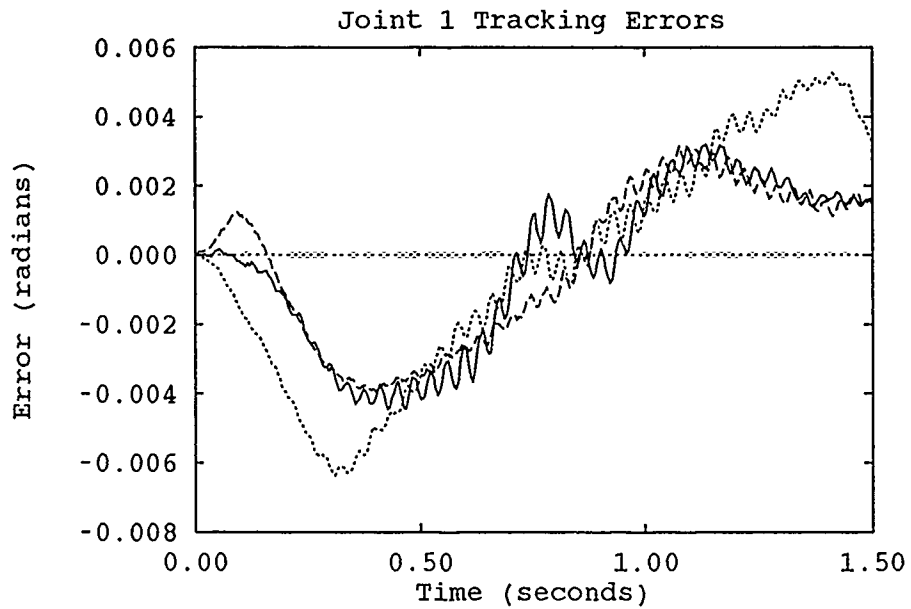


Figure J.22. Comparison of Initialized Runs After Learning - Trajectory 6

—	19 Parameters	13 Parameters
- - -	16 Parameters		

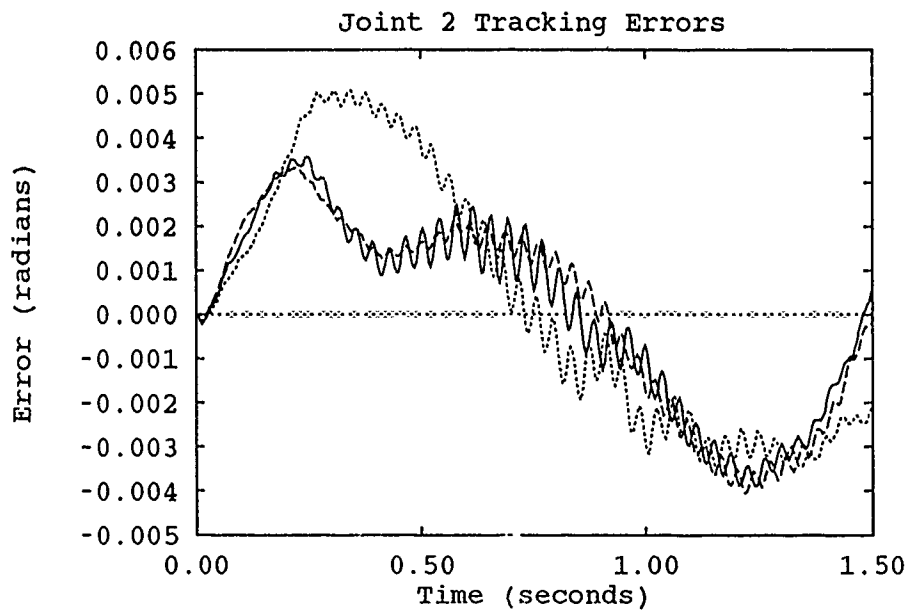


Figure J.23. Comparison of Initialized Runs After Learning - Trajectory 6

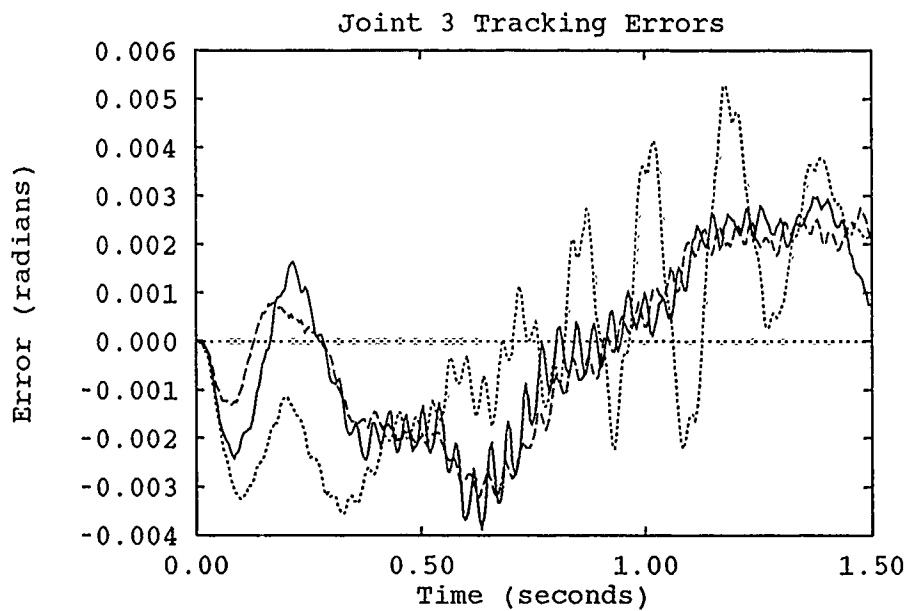


Figure J.24. Comparison of Initialized Runs After Learning - Trajectory 6

—	19 Parameters	13 Parameters
- - -	16 Parameters		

Appendix K. *Comparisons of 16- and 19-Parameter Learning Runs*

This section contains a comparison of the final learning runs for 16- and 19-parameter AMBC controllers for each test trajectory using uninitialized and initialized parameters.

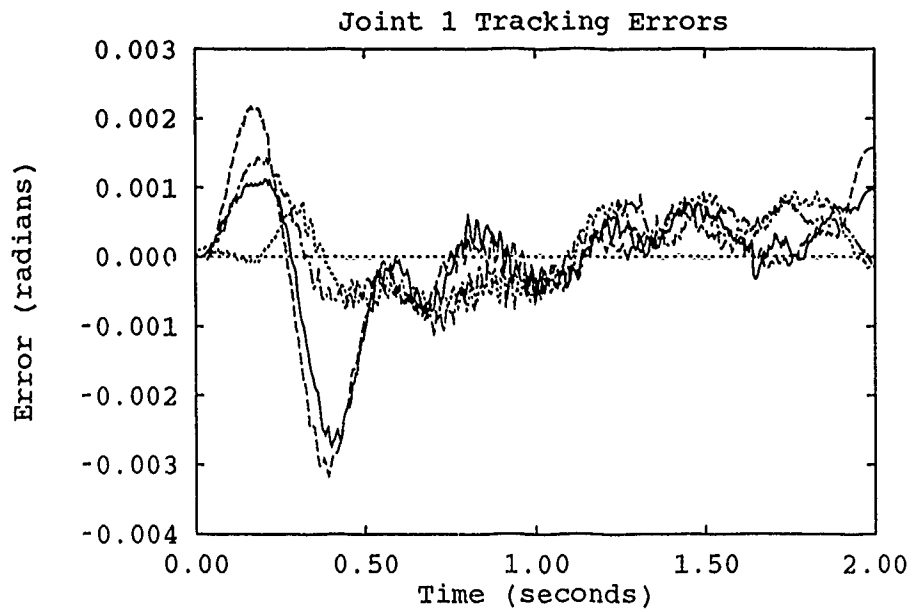


Figure K.1. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 2

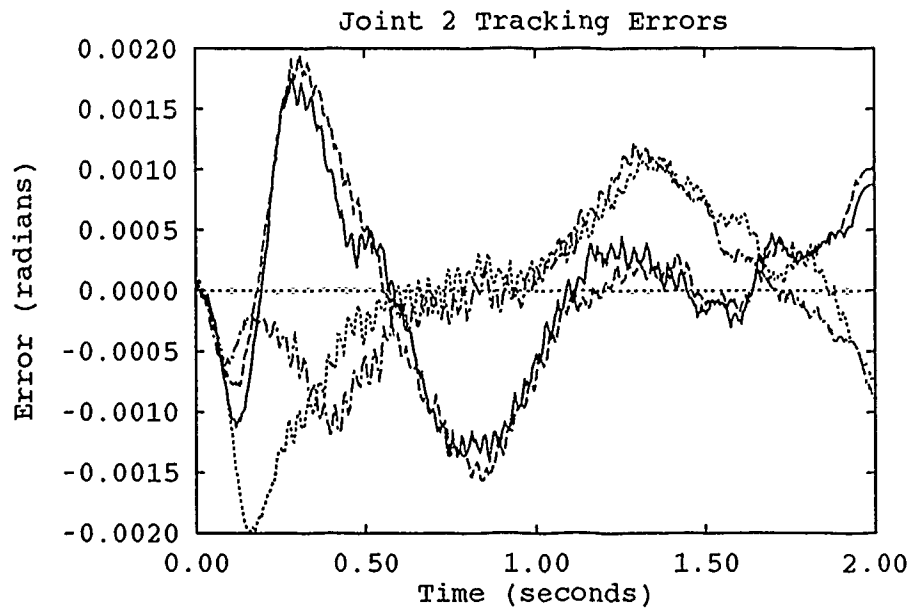


Figure K.2. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 2

—	19 Uninitialized Parameters	19 Initialized Parameters
---	16 Uninitialized Parameters	-.-.-.-	16 Initialized Parameters

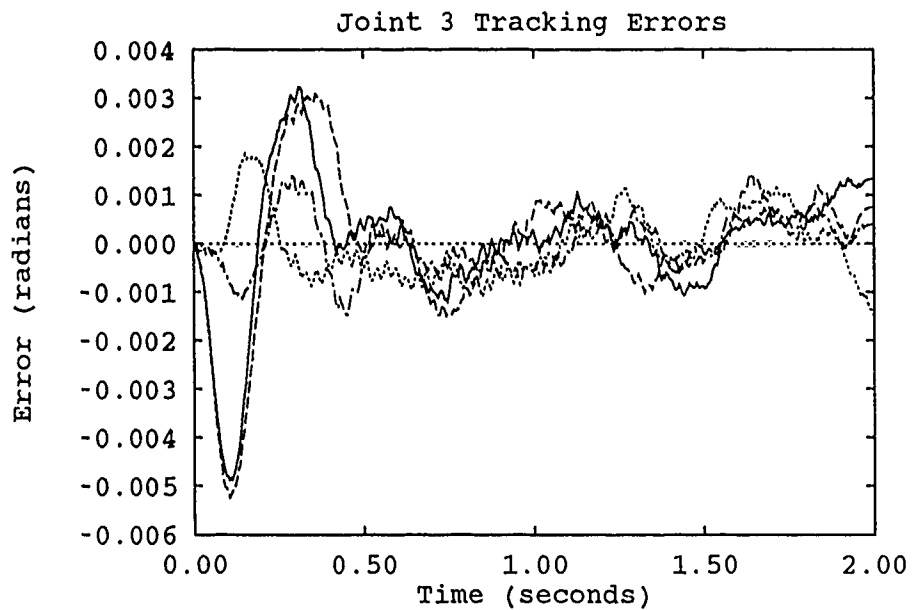


Figure K.3. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 2

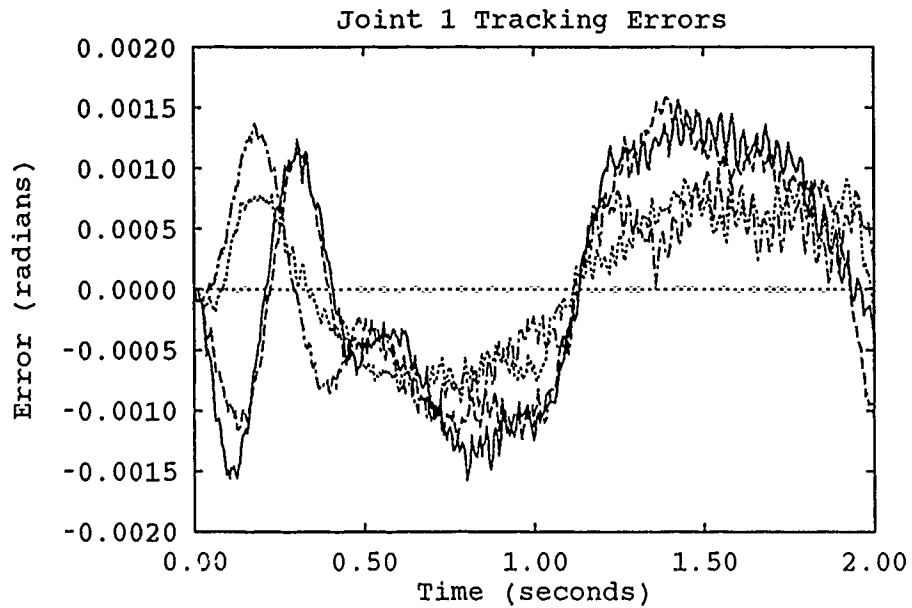


Figure K.4. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3

—	19 Uninitialized Parameters	19 Initialized Parameters
- - -	16 Uninitialized Parameters	- . - . -	16 Initialized Parameters

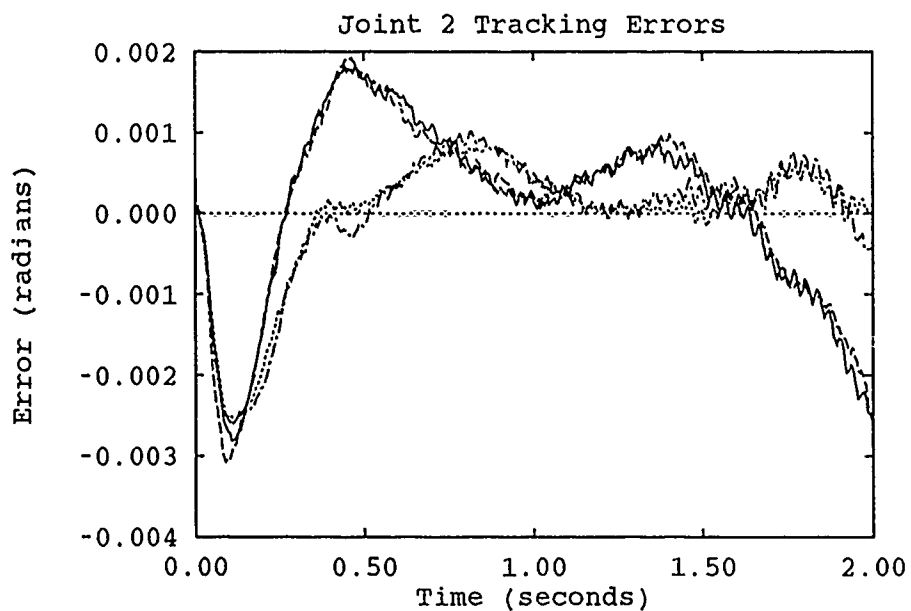


Figure K.5. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3

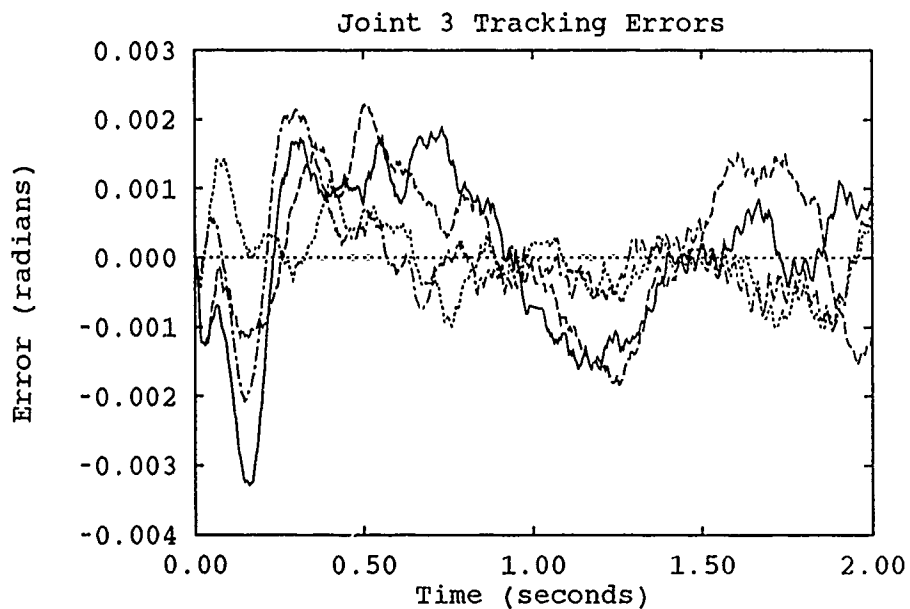


Figure K.6. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3

—	19 Uninitialized Parameters	19 Initialized Parameters
- - -	16 Uninitialized Parameters	- · - · -	16 Initialized Parameters

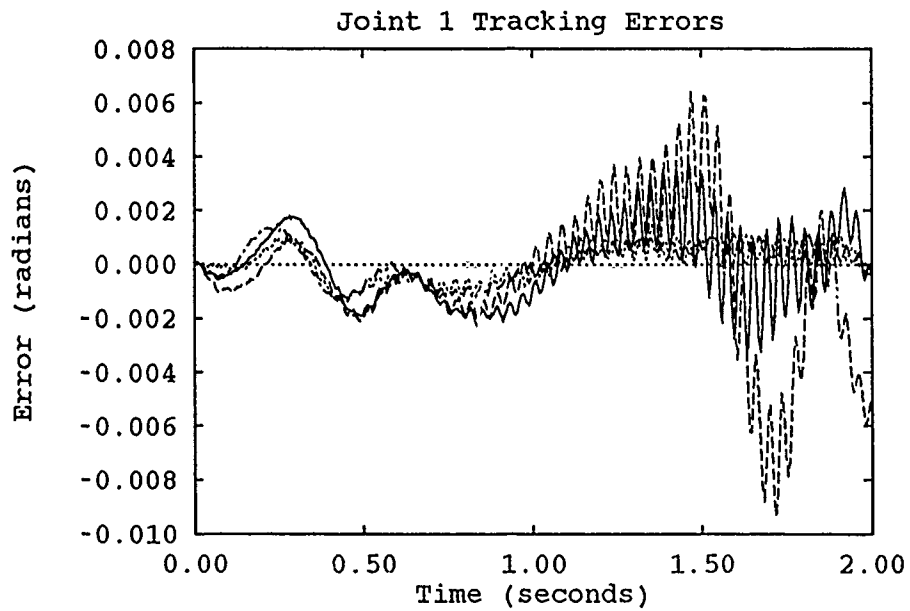


Figure K.7. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 3 w/ Payload

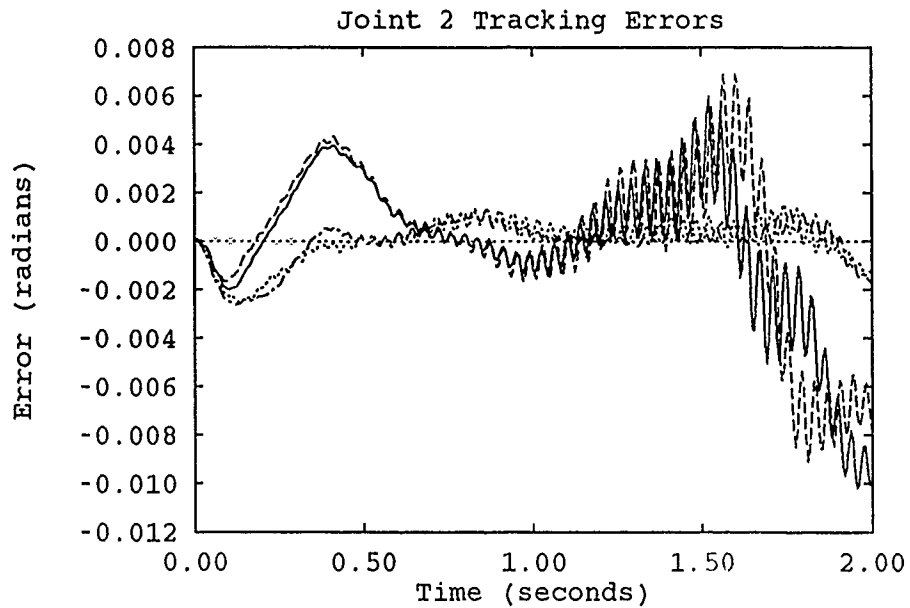


Figure K.8. Comparison of 16- and 19-Parameter Adaptation Runs Trajectory 3 w/ Payload

—	19 Uninitialized Parameters	19 Initialized Parameters
----	16 Uninitialized Parameters	-.-.-	16 Initialized Parameters

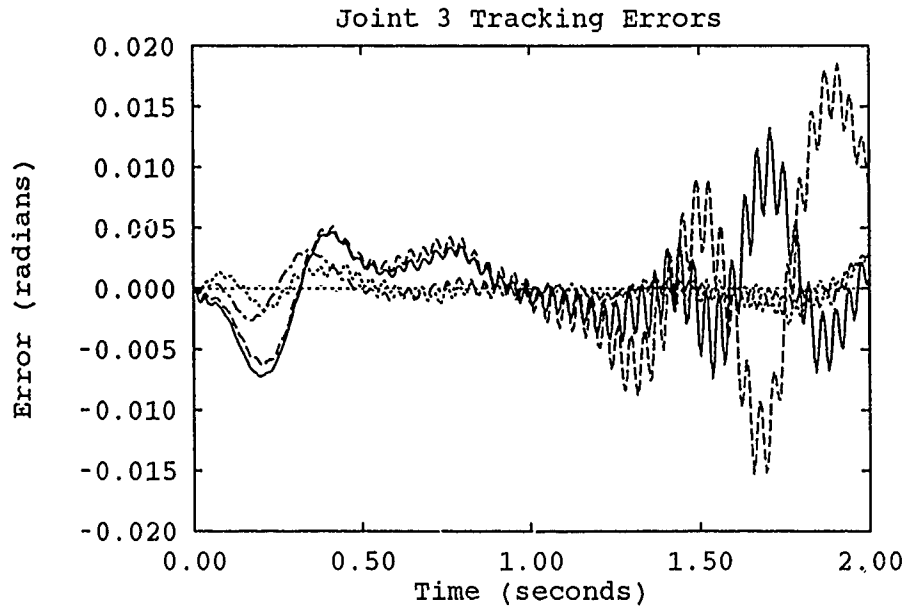


Figure K.9. Comparison of 10- and 19-Parameter Adaptation Runs - Trajectory 3 w/ Payload

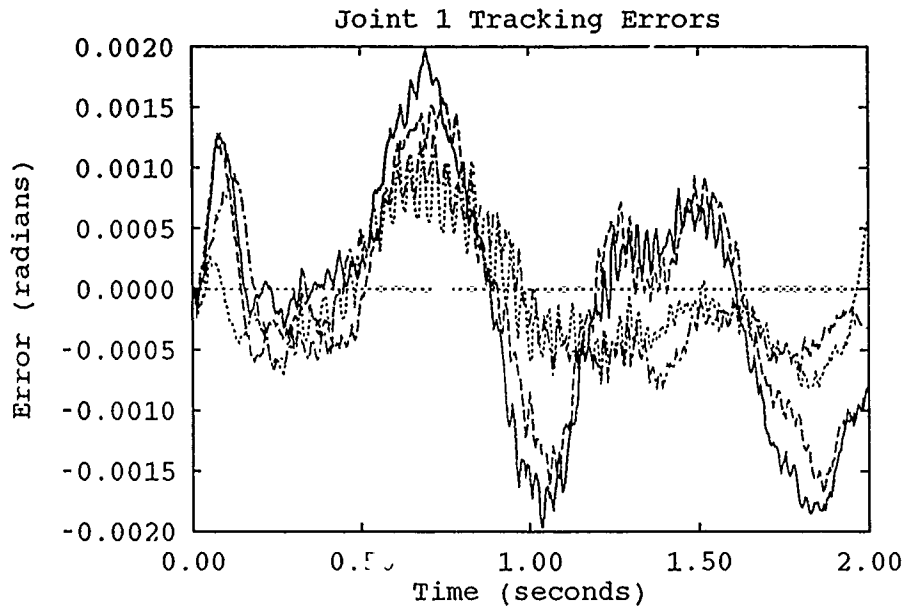


Figure K.10. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 4

—	19 Uninitialized Parameters	19 Initialized Parameters
----	16 Uninitialized Parameters	- - - -	16 Initialized Parameters

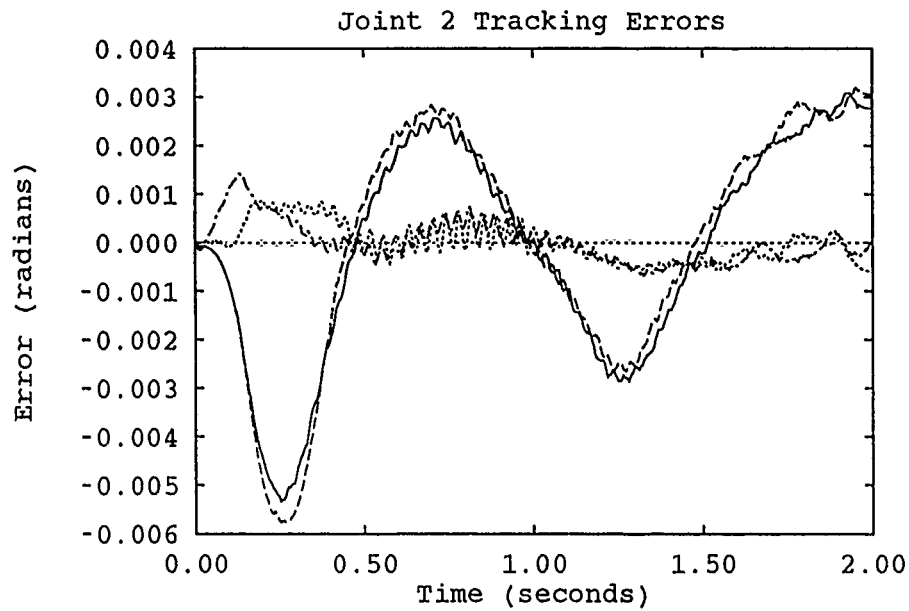


Figure K.11. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 4

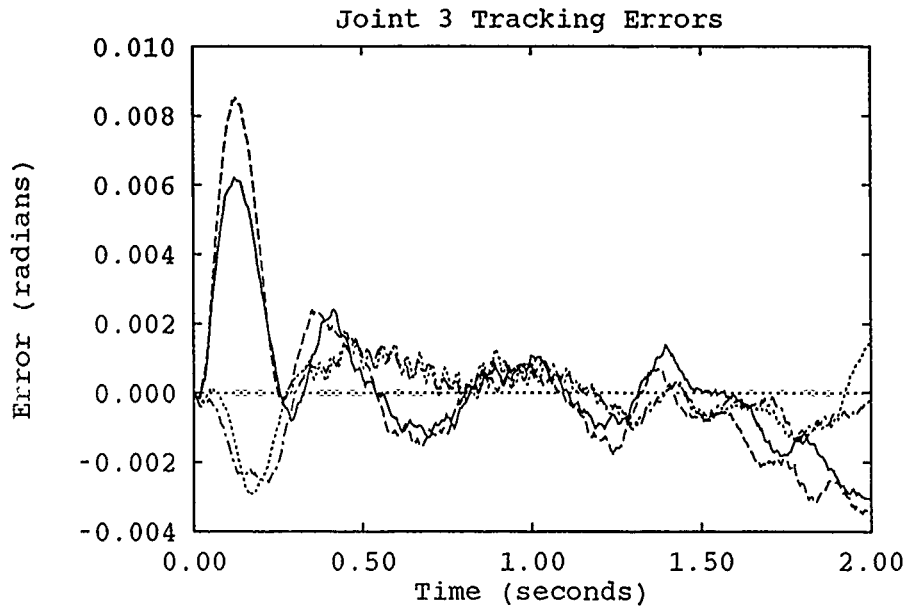


Figure K.12. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 4

—	19 Uninitialized Parameters	19 Initialized Parameters
----	16 Uninitialized Parameters	-.-.-.-	16 Initialized Parameters

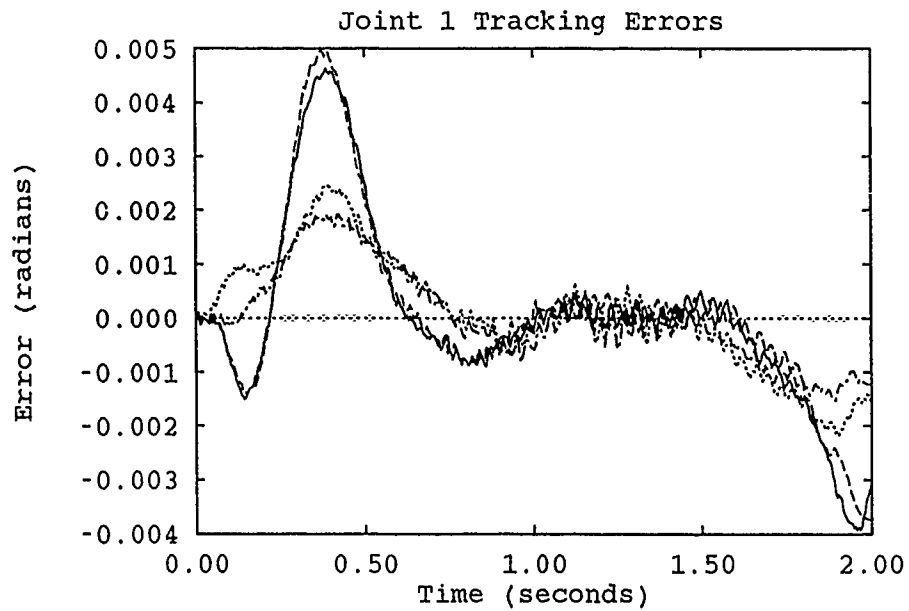


Figure K.13. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5

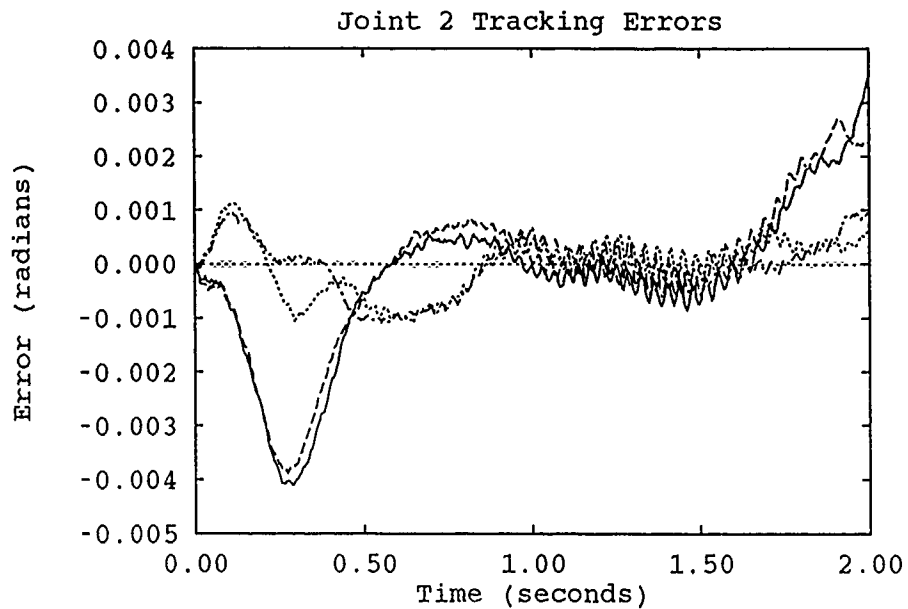


Figure K.14. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5

————	19 Uninitialized Parameters	19 Initialized Parameters
-----	16 Uninitialized Parameters	-.-.-.-	16 Initialized Parameters

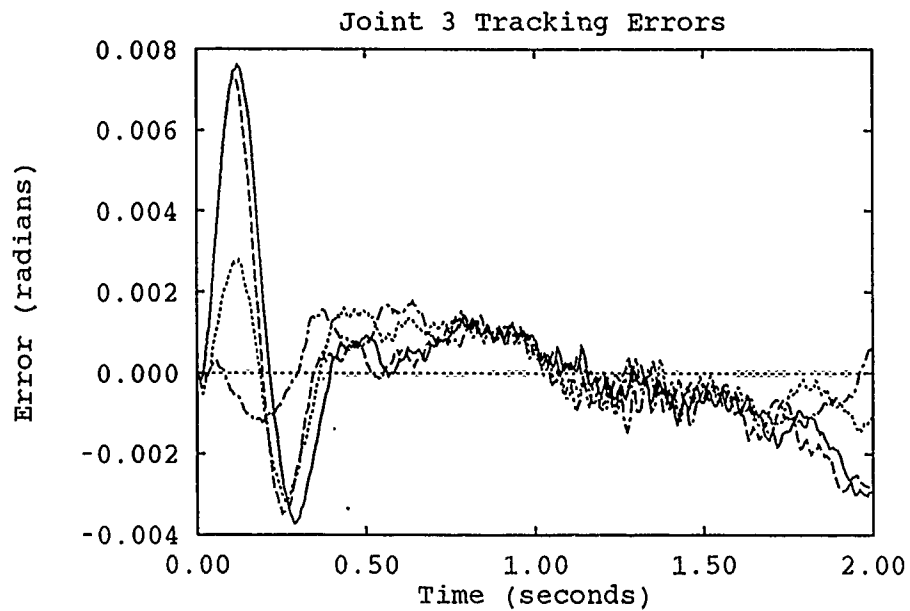


Figure K.15. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 5

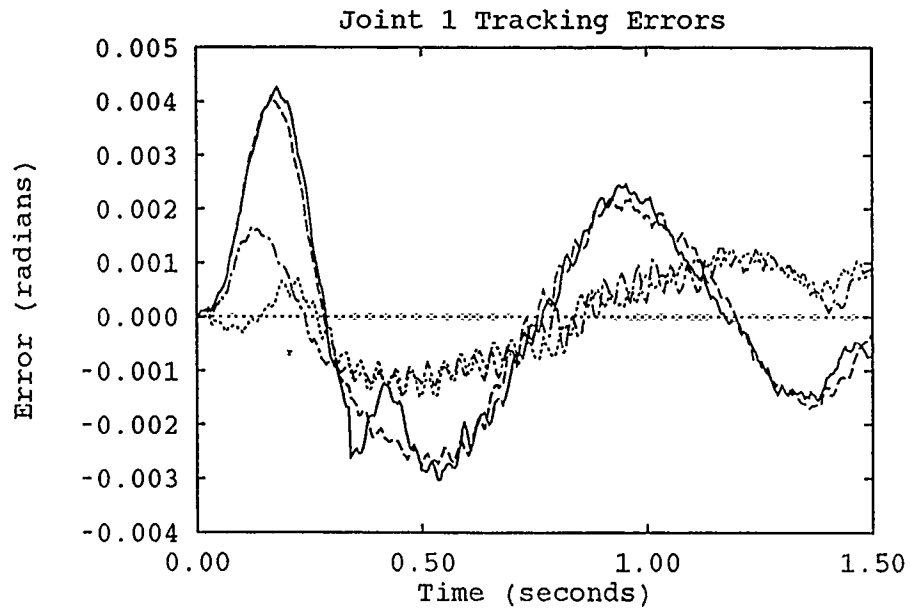


Figure K.16. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1

—	19 Uninitialized Parameters	19 Initialized Parameters
- - - -	16 Uninitialized Parameters	- . - . -	16 Initialized Parameters

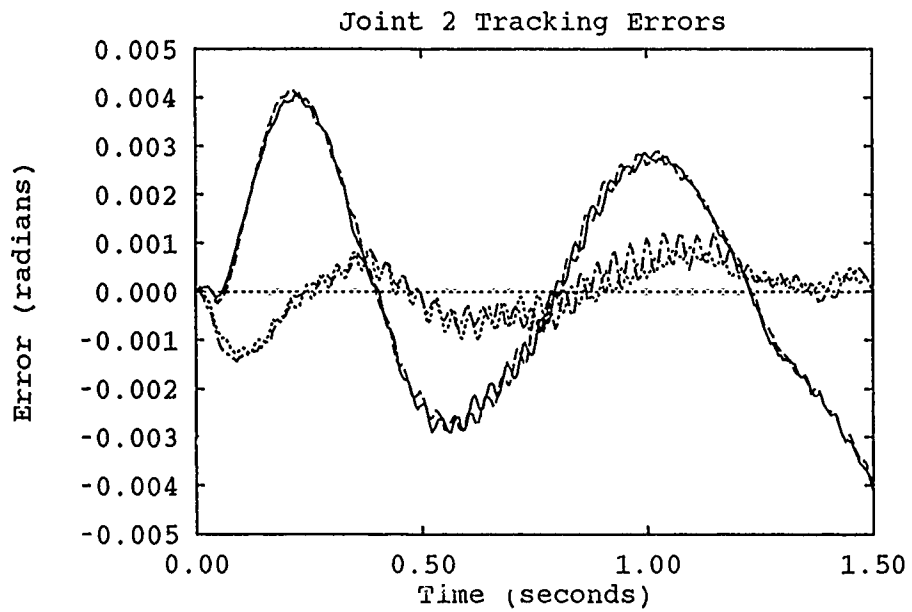


Figure K.17. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1

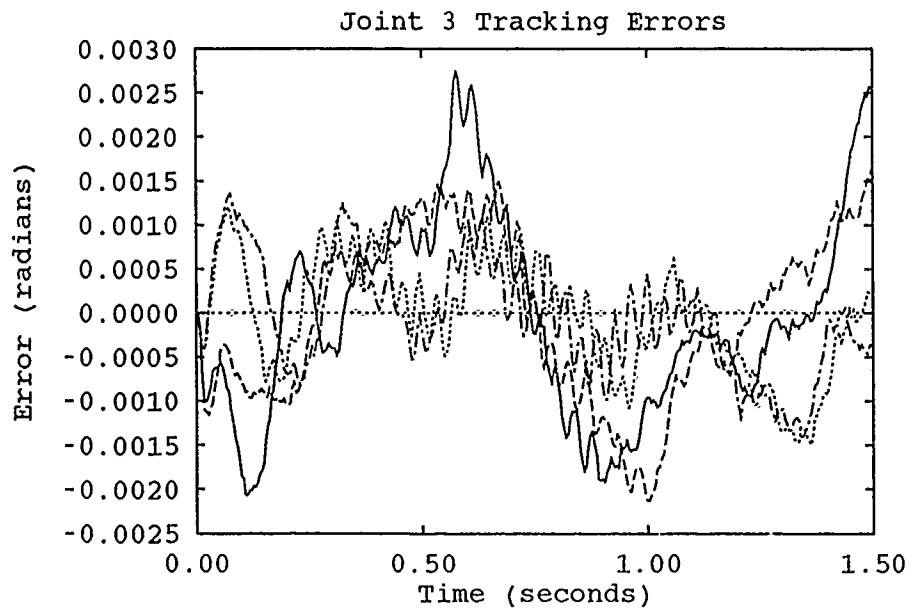


Figure K.18. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1

————	19 Uninitialized Parameters	19 Initialized Parameters
-----	16 Uninitialized Parameters	-.-.-.-	16 Initialized Parameters

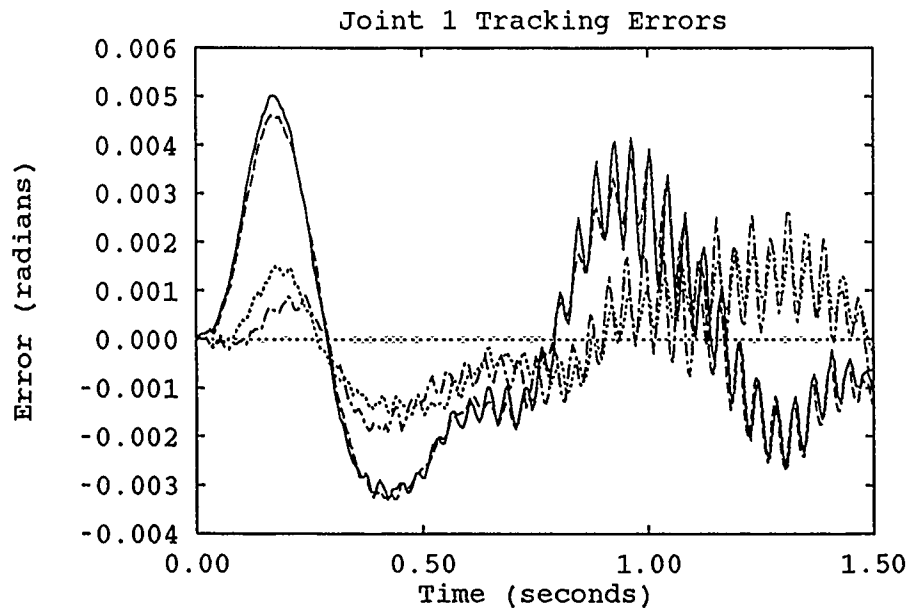


Figure K.19. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 w/ Payload

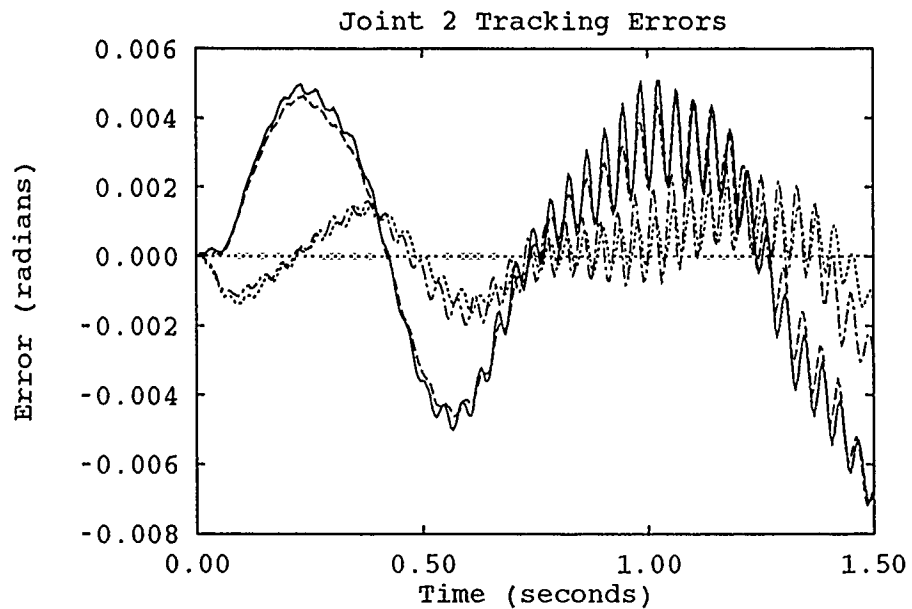


Figure K.20. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 w/ Payload

—	19 Uninitialized Parameters	19 Initialized Parameters
- - -	16 Uninitialized Parameters	- . - . -	16 Initialized Parameters

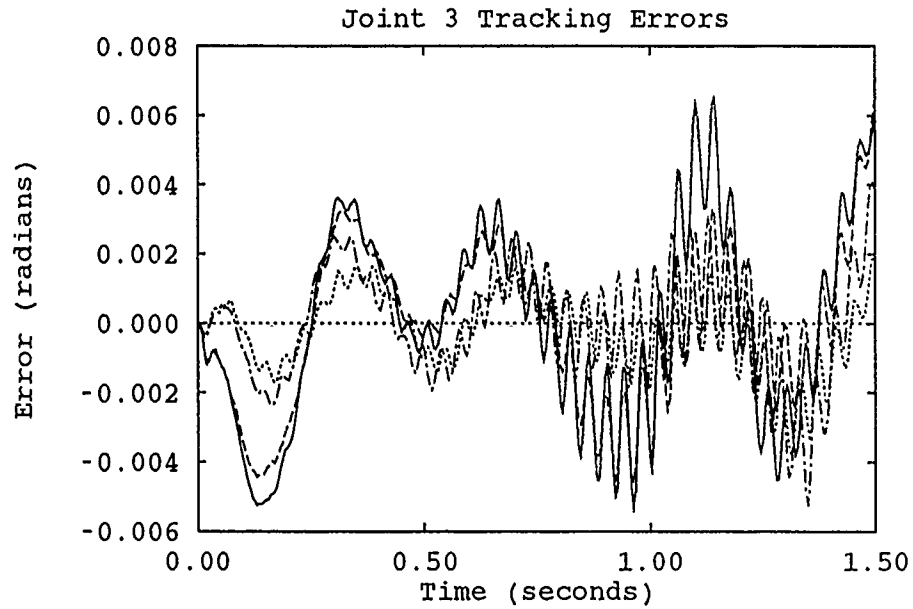


Figure K.21. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 1 w/ Payload

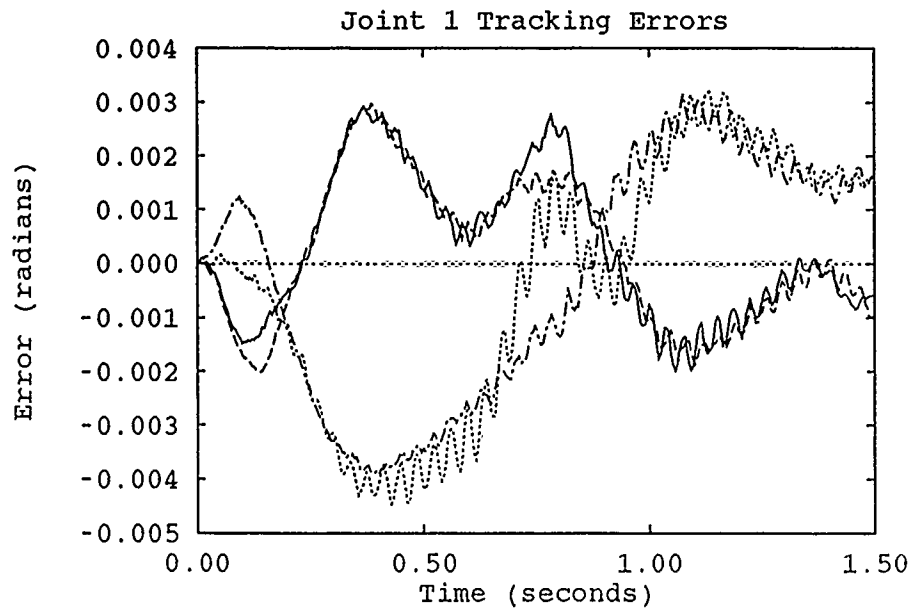


Figure K.22. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 6

————	19 Uninitialized Parameters	19 Initialized Parameters
-----	16 Uninitialized Parameters	-.-.-.-	16 Initialized Parameters

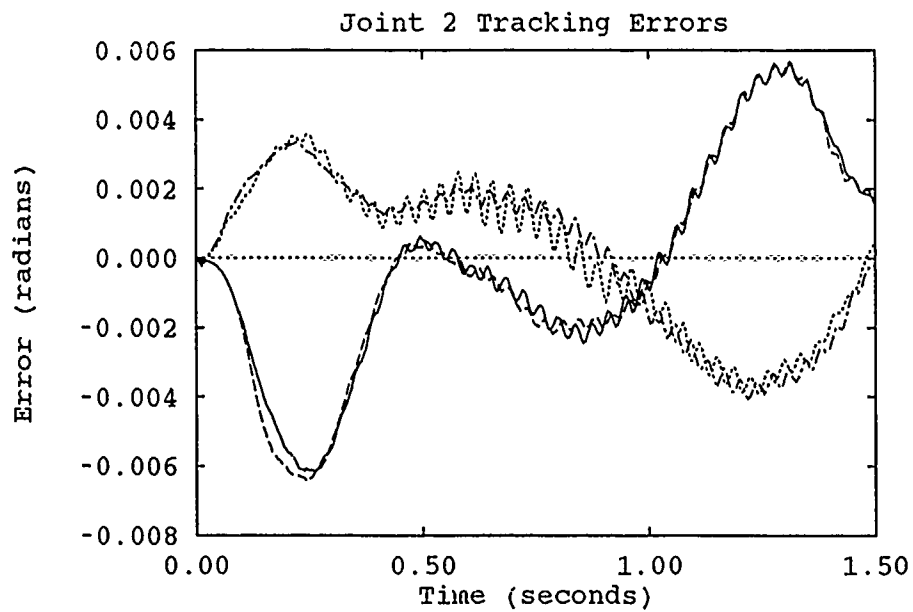


Figure K.23. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 6

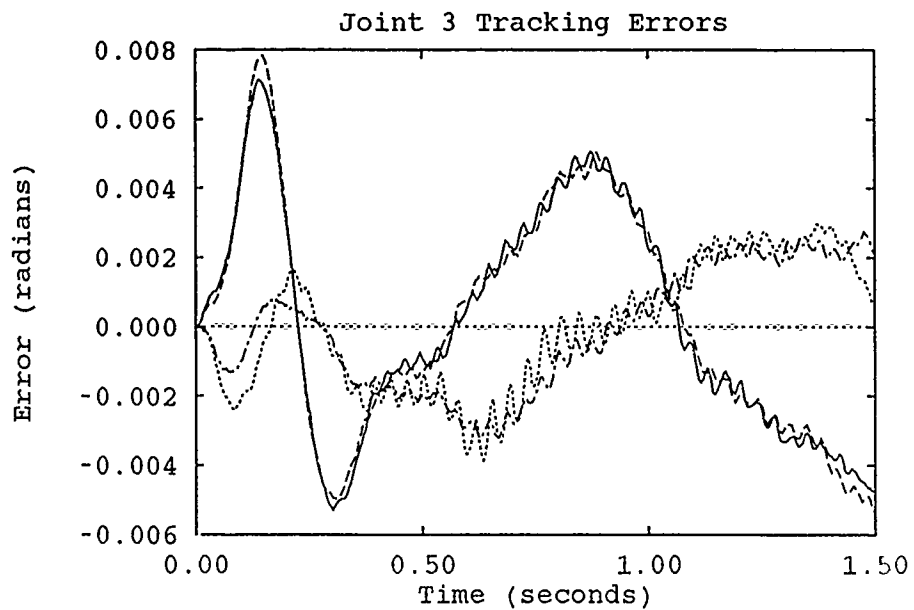


Figure K.24. Comparison of 16- and 19-Parameter Adaptation Runs - Trajectory 6

—	19 Uninitialized Parameters	19 Initialized Parameters
- - -	16 Uninitialized Parameters	- - - -	16 Initialized Parameters

Appendix L. *Learned Parameters for 13-Parameter Testing*

The results of conducting learning runs for all six test trajectories using 13 adaptive parameters are listed in the following tables. These trajectories were run with an assessment of all of the manipulator's parameters given (Initialized) and with all parameters set to zero (w/o Init). The learning process was conducted over a series of five runs for the initialized tests and nine runs for the uninitialized tests.

Table L.1. Learned 13-Parameter Values - Trajectory 1

Number	Trajectory 1 without payload			Trajectory 1 with payload		
	Nominal	Initialized	w/o Init	Nominal	Initialized	w/o Init
1	0.30	0.85	-4.96	0.68	1.37	-9.75
2	-52.09	-54.86	-30.73	-60.56	-63.99	-33.77
3	-7.53	-7.30	-10.12	-17.12	-18.59	-20.13
4	-0.01	-0.01	0.00	-0.01	-0.01	0.00
5	4.50	-1.73	-36.53	4.50	-2.58	-38.47
6	3.50	-0.23	-23.76	3.50	3.81	-19.08
7	3.50	1.09	-9.25	3.50	2.51	-10.02
8	5.95	7.62	10.42	5.95	7.24	9.95
9	6.82	10.58	20.56	6.82	8.78	18.74
10	3.91	4.30	4.72	3.91	1.98	2.82
11	1.00	0.21	-5.11	1.00	-0.59	-6.28
12	1.00	-0.34	-2.43	1.00	-0.95	-5.92
13	1.00	0.37	-5.90	1.00	-1.61	-11.04

Table L.2. Learned 13-Parameter Values - Trajectory 3

Number	Trajectory 3 without payload			Trajectory 3 with payload		
	Nominal	Initialized	w/o Init	Nominal	Initialized	w/o Init
1	0.30	-0.16	-3.28	0.68	-0.40	-7.44
2	-52.09	-52.32	-15.75	-60.56	-61.30	-19.53
3	-7.53	-8.34	-10.86	-17.12	-19.05	-24.65
4	-0.01	-0.01	0.00	-0.01	-0.01	0.00
5	4.50	6.03	-43.66	4.50	6.11	-50.12
6	3.50	5.91	-14.67	3.50	8.35	-6.52
7	3.50	2.16	-8.76	3.50	4.37	-13.15
8	5.95	7.29	7.81	5.95	6.24	3.30
9	6.82	7.79	32.64	6.82	5.23	31.98
10	3.91	4.47	6.84	3.91	3.65	1.98
11	1.00	0.69	-5.05	1.00	-0.30	-4.40
12	1.00	-1.03	1.75	1.00	-2.40	-4.07
13	1.00	0.00	-5.61	1.00	-2.58	-7.37

Table L.3. Learned 13-Parameter Values - Trajectories 2 and 4

Number	Nominal	Trajectory 2		Trajectory 4	
		Initialized	w/o Init	Initialized	w/o Init
1	0.30	0.26	-4.66	0.69	-4.18
2	-52.09	-51.67	-8.21	-52.26	-24.78
3	-7.53	-8.67	-9.23	-7.57	-7.37
4	-0.01	-0.01	0.00	-0.01	0.00
5	4.50	10.24	-47.31	0.51	-28.24
6	3.50	5.84	-5.37	4.07	-39.84
7	3.50	4.47	5.33	2.36	7.80
8	5.95	6.71	8.57	8.11	7.08
9	6.82	4.27	47.05	10.74	-2.69
10	3.91	5.32	9.90	4.75	7.43
11	1.00	0.76	-5.45	0.02	-0.99
12	1.00	-0.65	0.54	-0.15	-0.44
13	1.00	0.19	-2.92	0.25	-2.39

Table L.4. Learned 13-Parameter Values - Trajectories 5 and 6

		Trajectory 5		Trajectory 6	
Number	Nominal	Initialized	w/o Init	Initialized	w/o Init
1	0.30	1.25	6.85	2.29	4.86
2	-52.09	-54.46	-32.77	-46.72	-36.17
3	-7.53	-8.17	-11.51	-10.58	-6.63
4	-0.01	-0.01	0.00	-0.01	0.00
5	4.50	0.35	-24.52	17.40	-36.41
6	3.50	-1.55	-34.95	21.55	-43.15
7	3.50	4.62	11.56	6.24	-11.07
8	5.95	7.71	8.69	7.96	7.56
9	6.82	4.55	12.21	7.34	7.20
10	3.91	6.03	10.87	5.26	2.30
11	1.00	0.28	-2.44	0.55	-5.07
12	1.00	0.07	-0.27	-0.48	-5.30
13	1.00	0.00	-2.84	0.06	-7.07

Appendix M. *Learned Parameters for 16-Parameter Testing*

The results of conducting learning runs for all six test trajectories using adaptive 16-parameters are listed in the following tables. These trajectories were run with an assessment of all of the manipulator's parameters given (Initialized) and with all parameters set to zero (w/o Init). The learning process was conducted over a series of five runs for the initialized tests and nine runs for the uninitialized tests.

Table M.1. Learned 16-Parameter Values - Trajectory 1

Number	Trajectory 1 without payload			Trajectory 1 with payload		
	Nominal	Initialized	w/o Init	Nominal	Initialized	w/o Init
1	0.30	1.04	5.81	0.68	0.05	6.94
2	-52.09	-53.83	-17.64	-60.56	-62.62	-20.34
3	-7.53	-8.81	-8.46	-17.12	-18.86	-15.36
4	-0.01	-0.01	0.00	-0.01	-0.01	0.00
5	-0.03	0.23	-1.08	-0.06	0.34	-1.40
6	-1.57	-2.86	-3.34	-1.46	-3.40	-4.33
7	2.10	2.95	0.65	2.47	3.03	-0.71
8	-0.03	-0.70	-1.20	-0.07	-0.89	-1.51
9	0.67	2.99	11.41	1.51	2.99	11.84
10	-0.12	-0.23	-1.90	-0.26	-0.36	-2.65
11	4.50	4.81	7.04	4.50	5.45	7.49
12	3.50	4.08	3.58	3.50	4.37	4.32
13	3.50	3.45	3.18	3.50	3.51	3.07
14	5.95	7.46	5.02	5.95	6.91	4.61
15	6.82	9.51	17.00	6.82	8.50	18.29
16	3.91	5.86	10.14	3.91	5.67	8.30

Table M.2. Learned 16-Parameter Values - Trajectory 3

Number	Trajectory 3 without payload			Trajectory 3 with payload		
	Nominal	Initialized	w/o Init	Nominal	Initialized	w/o Init
1	0.30	-0.83	8.70	0.68	-0.90	8.65
2	-52.09	-52.29	-33.41	-60.56	-61.03	-39.21
3	-7.53	-8.85	-9.02	-17.12	-19.03	-21.75
4	-0.01	-0.01	0.00	-0.01	-0.01	0.00
5	-0.03	0.87	4.33	-0.06	0.83	3.31
6	-1.57	-2.26	-2.33	-1.46	-2.50	-3.73
7	2.10	2.13	6.43	2.47	2.46	7.08
8	-0.03	-0.65	1.69	-0.07	-0.79	1.79
9	0.67	0.91	8.19	1.51	1.47	11.79
10	-0.12	-0.05	-0.73	-0.26	-0.19	-0.95
11	4.50	5.26	5.21	4.50	5.39	7.39
12	3.50	4.15	3.81	3.50	4.33	5.16
13	3.50	3.55	3.26	3.50	3.70	5.34
14	5.95	7.84	10.11	5.95	7.23	7.52
15	6.82	6.88	20.25	6.82	6.97	21.92
16	3.91	4.48	11.24	3.91	4.66	8.20

Table M.3. Learned 16-Parameter Values - Trajectories 2 and 4

Number	Nominal	Trajectory 2		Trajectory 4	
		Initialized	w/o Init	Initialized	w/o Init
1	0.30	1.36	5.46	0.45	-3.57
2	-52.09	-51.35	-29.85	-50.99	-16.47
3	-7.53	-9.26	-17.22	-7.17	-3.74
4	-0.01	-0.01	0.00	-0.01	0.00
5	-0.03	-1.03	-1.03	0.20	3.34
6	-1.57	-1.35	-2.47	-0.86	-3.71
7	2.10	1.14	2.94	2.00	0.32
8	-0.03	0.11	1.20	-0.10	-1.42
9	0.67	0.46	6.39	2.11	10.51
10	-0.12	-0.37	-0.47	0.20	0.69
11	4.50	5.27	1.93	5.46	3.64
12	3.50	3.90	2.49	3.93	-1.95
13	3.50	3.56	0.24	2.86	1.82
14	5.95	7.36	2.27	8.13	9.70
15	6.82	6.12	21.36	8.86	-8.96
16	3.91	5.40	4.58	4.06	4.63

Table M.4. Learned 16-Parameter Values - Trajectories 5 and 6

		Trajectory 5		Trajectory 6	
Number	Nominal	Initialized	w/o Init	Initialized	w/o Init
1	0.30	0.40	-6.43	-0.29	-0.56
2	-52.09	-53.11	-21.81	-44.49	-29.35
3	-7.53	-8.14	-13.48	-9.92	6.23
4	-0.01	-0.01	0.00	-0.01	0.00
5	-0.03	0.78	1.02	0.24	2.93
6	-1.57	-1.43	-1.75	-0.47	2.08
7	2.10	1.36	-6.85	-2.94	24.57
8	-0.03	-0.03	-0.27	0.02	1.04
9	0.67	3.65	15.35	-1.58	14.23
10	-0.12	-0.14	-0.03	-0.71	3.05
11	4.50	5.10	4.18	7.29	4.73
12	3.50	3.54	1.58	4.55	-0.97
13	3.50	3.45	-0.43	3.68	5.46
14	5.95	7.00	3.99	6.96	10.59
15	6.82	5.70	13.34	9.99	-3.93
16	3.91	5.73	5.18	6.08	-4.40

Appendix N. *Learned Parameters for 19-Parameter Testing*

The results of conducting learning runs for all six test trajectories using adaptive 19-parameters are listed in the following tables. These trajectories were run with an assessment of all of the manipulator's parameters given (Initialized) and with all parameters set to zero (w/o Init). The learning process was conducted over a series of five runs for the initialized tests and nine runs for the uninitialized tests.

Table N.1. Learned 19-Parameter Values - Trajectory 1

Number	Trajectory 1 without payload			Trajectory 1 with payload		
	Nominal	Initialized	w/o Init	Nominal	Initialized	w/o Init
1	0.30	0.87	6.78	0.68	1.31	6.71
2	-52.09	-53.15	-17.88	-60.56	-62.80	-20.81
3	-7.53	-9.63	-8.88	-17.12	-18.73	-15.39
4	-0.01	-0.01	0.00	-0.01	-0.01	0.00
5	-0.03	0.38	-1.18	-0.06	0.11	-1.49
6	-1.57	-2.77	-3.36	-1.46	-3.45	-4.41
7	2.10	3.40	0.82	2.47	3.59	-0.65
8	-0.03	-0.65	-1.24	-0.07	-0.91	-1.57
9	0.67	3.04	11.58	1.51	3.59	11.69
10	-0.12	-0.16	-1.80	-0.26	-0.25	-2.62
11	4.50	4.23	7.00	4.50	5.37	7.57
12	3.50	4.16	3.61	3.50	4.54	4.44
13	3.50	3.52	3.33	3.50	3.65	3.20
14	5.95	7.35	4.74	5.95	7.13	4.54
15	6.82	9.88	17.13	6.82	8.73	18.04
16	3.91	6.02	9.45	3.91	5.12	8.21
17	1.00	1.64	1.22	1.00	-0.61	0.27
18	1.00	0.87	-0.16	1.00	0.53	-0.45
19	1.00	0.91	-0.33	1.00	1.10	-0.65

Table N.2. Learned 19-Parameter Values - Trajectory 3

Number	Trajectory 3 without payload			Trajectory 3 with payload		
	Nominal	Initialized	w/o Init	Nominal	Initialized	w/o Init
1	0.30	-1.22	9.24	0.68	-1.24	8.13
2	-52.09	-51.47	-33.83	-60.56	-60.75	-38.48
3	-7.53	-7.94	-10.44	-17.12	-19.23	-18.18
4	-0.01	-0.01	0.00	-0.01	-0.01	0.00
5	-0.03	1.19	3.96	-0.06	0.70	4.38
6	-1.57	-2.36	-2.09	-1.46	-3.12	-4.25
7	2.10	2.38	6.26	2.47	2.56	6.79
8	-0.03	-0.64	1.66	-0.07	-0.98	1.59
9	0.67	0.64	8.36	1.51	0.86	10.26
10	-0.12	-0.04	-0.51	-0.26	-0.23	-1.03
11	4.50	5.10	5.15	4.50	5.70	6.33
12	3.50	4.30	3.77	3.50	4.70	5.16
13	3.50	3.56	3.32	3.50	4.11	4.62
14	5.95	7.61	10.07	5.95	7.11	9.09
15	6.82	6.42	20.54	6.82	7.01	21.46
16	3.91	4.65	10.67	3.91	4.88	9.54
17	1.00	0.44	-0.63	1.00	0.16	-2.74
18	1.00	0.45	-0.21	1.00	0.34	-1.07
19	1.00	0.80	0.01	1.00	0.50	-0.83

Table N.3. Learned 19-Parameter Values - Trajectories 2 and 4

Number	Nominal	Trajectory 2		Trajectory 4	
		Initialized	w/o Init	Initialized	w/o Init
1	0.30	0.52	6.90	1.30	-3.62
2	-52.09	-51.19	-29.88	-50.89	-16.61
3	-7.53	-9.03	-16.96	-7.79	-4.48
4	-0.01	-0.01	0.00	-0.01	0.00
5	-0.03	-0.73	-1.18	-0.05	3.47
6	-1.57	-1.80	-2.24	-0.44	-3.44
7	2.10	1.40	2.93	2.82	0.41
8	-0.03	0.21	1.13	-0.06	-1.56
9	0.67	0.32	6.64	1.83	10.43
10	-0.12	-0.48	-0.47	0.20	0.68
11	4.50	5.05	2.09	5.46	3.80
12	3.50	4.17	2.72	4.00	-1.90
13	3.50	3.08	0.17	2.69	1.56
14	5.95	7.78	1.79	8.42	9.48
15	6.82	5.93	21.30	7.60	-9.34
16	3.91	5.70	4.89	4.14	4.43
17	1.00	1.93	3.96	1.22	0.70
18	1.00	0.69	0.49	0.75	-0.19
19	1.00	0.89	-0.42	0.96	-0.38

Table N.4. Learned 19-Parameter Values - Trajectories 5 and 6

Number	Nominal	Trajectory 5		Trajectory 6	
		Initialized	w/o Init	Initialized	w/o Init
1	0.30	0.81	-6.71	0.88	-0.52
2	-52.09	-53.37	-21.84	-42.77	-29.18
3	-7.53	-8.08	-13.66	-0.50	6.65
4	-0.01	-0.01	0.00	-0.01	0.00
5	-0.03	0.73	1.09	-5.17	2.94
6	-1.57	-1.44	-1.75	-0.41	1.98
7	2.10	0.18	-6.83	-2.47	24.47
8	-0.03	0.03	-0.21	0.03	1.10
9	0.67	3.85	15.60	7.18	14.06
10	-0.12	-0.11	-0.08	-0.69	3.08
11	4.50	4.85	4.56	3.43	4.88
12	3.50	3.62	1.64	4.75	-0.92
13	3.50	3.44	-0.33	9.66	5.53
14	5.95	6.97	3.50	6.01	10.61
15	6.82	4.91	12.94	1.73	-3.96
16	3.91	5.50	5.00	6.12	-4.24
17	1.00	2.22	4.50	0.73	-0.78
18	1.00	0.59	0.01	0.70	-0.20
19	1.00	0.68	-0.08	0.01	-0.70

Appendix O. Comparison of Learned Parameters

The results of conducting learning runs for all six test trajectories using 13, 16, and 19 adaptive parameters are listed in the following tables. These trajectories were run with an assessment of all of the manipulator's parameters given (Initialized) and with all parameters set to zero (w/o Init). The learning process was conducted over a series of five runs for the initialized tests and nine runs for the uninitialized tests. The last nine parameters in the 13-parameter adaptation are placed so that they coincide with the same friction parameters in the 16- and 19-parameter configurations.

Table O.1. Learned Parameter Values - Trajectory 1

Number	Nominal	Trajectory 1 Initialized			Trajectory 1 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.30	0.85	1.04	0.87	-4.96	5.81	6.78
2	-52.09	-54.86	-53.83	-53.15	-30.73	-17.64	-17.88
3	-7.53	-7.30	-8.81	-9.63	-10.12	-8.46	-8.88
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.03		0.23	0.38		-1.08	-1.18
6	-1.57		-2.86	-2.77		-3.34	-3.36
7	2.10		2.95	3.40		0.65	0.82
8	-0.03		-0.70	-0.65		-1.20	-1.24
9	0.67		2.99	3.04		11.41	11.58
10	-0.12		-0.23	-0.16		-1.90	-1.80
11	4.50	-1.73	4.81	4.23	-36.53	7.04	7.00
12	3.50	-0.23	4.08	4.16	-23.76	3.58	3.61
13	3.50	1.09	3.45	3.52	-9.25	3.18	3.33
14	5.95	7.62	7.46	7.35	10.42	5.02	4.74
15	6.82	10.58	9.51	9.88	20.56	17.00	17.13
16	3.91	4.30	5.86	6.02	4.72	10.14	9.45
17	1.00	0.21		1.64	-5.11		1.22
18	1.00	-0.34		0.87	-2.43		-0.16
19	1.00	0.37		0.91	-5.90		-0.33

Table O.2. Learned Parameter Values - Trajectory 1 with payload

Number	Nominal	Trajectory 1 Initialized			Trajectory 1 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.68	1.37	0.05	1.31	-9.75	6.94	6.71
2	-60.56	-63.99	-62.62	-62.80	-33.77	-20.34	-20.81
3	-17.12	-18.59	-18.86	-18.73	-20.13	-15.36	-15.39
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.06		0.34	0.11		-1.40	-1.49
6	-1.46		-3.40	-3.45		-4.33	-4.41
7	2.47		3.03	3.59		-0.71	-0.65
8	-0.07		-0.89	-0.91		-1.51	-1.57
9	1.51		2.99	3.59		11.84	11.69
10	-0.26		-0.36	-0.25		-2.65	-2.62
11	4.50	-2.58	5.45	5.37	-38.47	7.49	7.57
12	3.50	3.81	4.37	4.54	-19.08	4.32	4.44
13	3.50	2.51	3.51	3.65	-10.02	3.07	3.20
14	5.95	7.24	6.91	7.13	9.95	4.61	4.54
15	6.82	8.78	8.50	8.73	18.74	18.29	18.04
16	3.91	1.98	5.67	5.12	2.82	8.30	8.21
17	1.00	-0.59		-0.61	-6.28		0.27
18	1.00	-0.95		0.53	-5.92		-0.45
19	1.00	-1.61		1.10	-11.04		-0.65

Table O.3. Learned Parameter Values - Trajectory 2

Number	Nominal	Trajectory 2 Initialized			Trajectory 2 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.30	0.26	1.36	0.52	-4.66	5.46	6.90
2	-52.09	-51.67	-51.35	-51.19	-8.21	-29.85	-29.88
3	-7.53	-8.67	-9.26	-9.03	-9.23	-17.22	-16.96
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.03		-1.03	-0.73		-1.03	-1.18
6	-1.57		-1.35	-1.80		-2.47	-2.24
7	2.10		1.14	1.40		2.94	2.93
8	-0.03		0.11	0.21		1.20	1.13
9	0.67		0.46	0.32		6.39	6.64
10	-0.12		-0.37	-0.48		-0.47	-0.47
11	4.50	10.24	5.27	5.05	-47.31	1.93	2.09
12	3.50	5.84	3.90	4.17	-5.37	2.49	2.72
13	3.50	4.47	3.56	3.08	5.33	0.24	0.17
14	5.95	6.71	7.36	7.78	8.57	2.27	1.79
15	6.82	4.27	6.12	5.93	47.05	21.36	21.30
16	3.91	5.32	5.40	5.70	9.90	4.58	4.89
17	1.00	0.76		1.93	-5.45		3.96
18	1.00	-0.65		0.69	0.54		0.49
19	1.00	0.19		0.89	-2.92		-0.42

Table O.4. Learned Parameter Values - Trajectory 3

Number	Nominal	Trajectory 3 Initialized			Trajectory 3 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.30	-0.16	-0.83	-1.22	-3.28	8.70	9.24
2	-52.09	-52.32	-52.29	-51.47	-15.75	-33.41	-33.83
3	-7.53	-8.34	-8.85	-7.94	-10.86	-9.02	-10.44
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.03		0.87	1.19		4.33	3.96
6	-1.57		-2.26	-2.36		-2.33	-2.09
7	2.10		2.13	2.38		6.43	6.26
8	-0.03		-0.65	-0.64		1.69	1.66
9	0.67		0.91	0.64		8.19	8.36
10	-0.12		-0.05	-0.04		-0.73	-0.51
11	4.50	6.03	5.26	5.10	-43.66	5.21	5.15
12	3.50	5.91	4.15	4.30	-14.67	3.81	3.77
13	3.50	2.16	3.55	3.56	-8.76	3.26	3.32
14	5.95	7.29	7.84	7.61	7.81	10.11	10.07
15	6.82	7.79	6.88	6.42	32.64	20.25	20.54
16	3.91	4.47	4.48	4.65	6.84	11.24	10.67
17	1.00	0.69		0.44	-5.05		-0.63
18	1.00	-1.03		0.45	1.75		-0.21
19	1.00	0.00		0.80	-5.61		0.01

Table O.5. Learned Parameter Values - Trajectory 3 with payload

Number	Nominal	Trajectory 3 Initialized			Trajectory 3 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.68	-0.40	-0.90	-1.24	-7.44	8.65	8.13
2	-60.56	-61.30	-61.03	-60.75	-19.53	-39.21	-38.48
3	-17.12	-19.05	-19.03	-19.23	-24.65	-21.75	-18.18
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.06		0.83	0.70		3.31	4.38
6	-1.46		-2.50	-3.12		-3.73	-4.25
7	2.47		2.46	2.56		7.08	6.79
8	-0.07		-0.79	-0.98		1.79	1.59
9	1.51		1.47	0.86		11.79	10.26
10	-0.26		-0.19	-0.23		-0.95	-1.03
11	4.50	6.11	5.39	5.70	-50.12	7.39	6.33
12	3.50	8.35	4.33	4.70	-6.52	5.16	5.16
13	3.50	4.37	3.70	4.11	-13.15	5.34	4.62
14	5.95	6.24	7.23	7.11	3.30	7.52	9.09
15	6.82	5.23	6.97	7.01	31.98	21.92	21.46
16	3.91	3.65	4.66	4.88	1.98	8.20	9.54
17	1.00	-0.30		0.16	-4.40		-2.74
18	1.00	-2.40		0.34	-4.07		-1.07
19	1.00	-2.58		0.50	-7.37		-0.83

Table O.6. Learned Parameter Values - Trajectory 4

Number	Nominal	Trajectory 4 Initialized			Trajectory 4 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.30	0.69	0.45	1.30	-4.18	-3.57	-3.62
2	-52.09	-52.26	-50.99	-50.89	-24.78	-16.47	-16.61
3	-7.53	-7.57	-7.17	-7.79	-7.37	-3.74	-4.48
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.03		0.20	-0.05		3.34	3.47
6	-1.57		-0.86	-0.44		-3.71	-3.44
7	2.10		2.00	2.82		0.32	0.41
8	-0.03		-0.10	-0.06		-1.42	-1.56
9	0.67		2.11	1.83		10.51	10.43
10	-0.12		0.20	0.20		0.69	0.68
11	4.50	0.51	5.46	5.46	-28.24	3.64	3.80
12	3.50	4.07	3.93	4.00	-39.84	-1.95	-1.90
13	3.50	2.36	2.86	2.69	7.80	1.82	1.56
14	5.95	8.11	8.13	8.42	7.08	9.70	9.48
15	6.82	10.74	8.86	7.60	-2.69	-8.96	-9.34
16	3.91	4.75	4.06	4.14	7.43	4.63	4.43
17	1 00	0.02		1.22	-0.99		0.70
18	1.00	-0.15		0.75	-0.44		-0.19
19	1.00	0.25		0.96	-2.39		-0.38

Table O.7. Learned Parameter Values - Trajectory 5

Number	Nominal	Trajectory 5 Initialized			Trajectory 5 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.30	1.25	0.40	0.81	6.85	-6.43	-6.71
2	-52.09	-54.46	-53.11	-53.37	-32.77	-21.18	-21.84
3	-7.53	-8.17	-8.14	-8.08	-11.51	-13.48	-13.66
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.03		0.78	0.73		1.02	1.09
6	-1.57		-1.43	-1.44		-1.75	-1.75
7	2.10		1.36	0.18		-6.85	-6.83
8	-0.03		-0.03	0.03		-0.27	-0.21
9	0.67		3.65	3.85		15.35	15.60
10	-0.12		-0.14	-0.11		-0.03	-0.08
11	4.50	0.35	5.10	4.85	-24.52	4.18	4.56
12	3.50	-1.55	3.54	3.62	-34.95	1.58	1.64
13	3.50	4.62	3.45	3.44	11.56	-0.43	-0.33
14	5.95	7.71	7.00	6.97	8.69	3.99	3.50
15	6.82	4.55	5.70	4.91	12.21	13.34	12.94
16	3.91	6.03	5.73	5.50	10.87	5.18	5.00
17	1.00	0.28		2.22	-2.44		4.50
18	1.00	0.07		0.59	-0.27		0.01
19	1.00	0.00		0.68	-2.82		-0.08

Table O.8. Learned Parameter Values - Trajectory 6

Number	Nominal	Trajectory 6 Initialized			Trajectory 6 Uninitialized		
		13 par	16 par	19 par	13 par	16 par	19 par
1	0.30	2.29	-0.29	0.88	4.86	-0.56	-0.52
2	-52.09	-46.72	-44.49	-42.77	-36.17	-29.35	-29.18
3	-7.53	-10.58	-9.92	-0.50	-6.63	6.23	6.65
4	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00
5	-0.03		0.24	-5.17		2.93	2.94
6	-1.57		-0.47	-0.41		2.08	1.98
7	2.10		-2.94	-2.47		24.57	24.47
8	-0.03		0.02	0.03		1.04	1.10
9	0.67		-1.58	7.18		14.23	14.06
10	-0.12		-0.71	-0.69		3.05	3.08
11	4.50	17.40	7.29	3.43	-36.41	4.73	4.88
12	3.50	21.55	4.55	4.75	-43.15	-0.97	-0.92
13	3.50	6.24	3.68	9.66	-11.07	5.46	5.53
14	5.95	7.96	6.96	6.01	7.56	10.59	10.61
15	6.82	7.34	9.99	1.73	7.20	-3.93	-3.96
16	3.91	5.26	6.08	6.12	2.30	-4.40	-4.24
17	1.00	0.55		0.73	-5.07		-0.78
18	1.00	-0.48		0.70	-5.30		-0.20
19	1.00	0.06		0.01	-7.07		-0.70

Appendix P. *Tarokh Algorithm Runs*

This section contains plots comparing the errors produced on Trajectory 0 versus other test trajectories using the set of design parameters chosen for Trajectory 0.

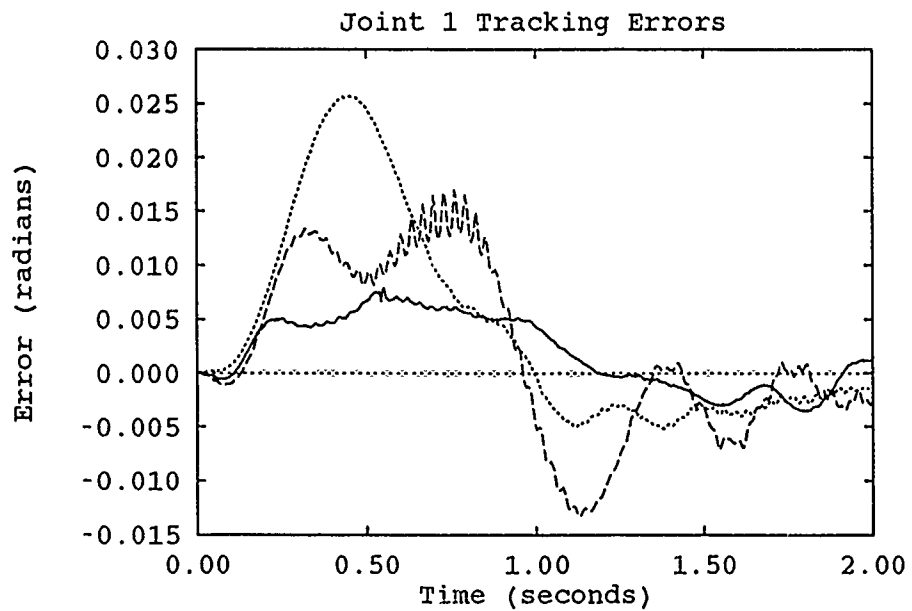


Figure P.1. The Effects of Trajectory 0 Tuning Parameters on Trajectories 2 and 3

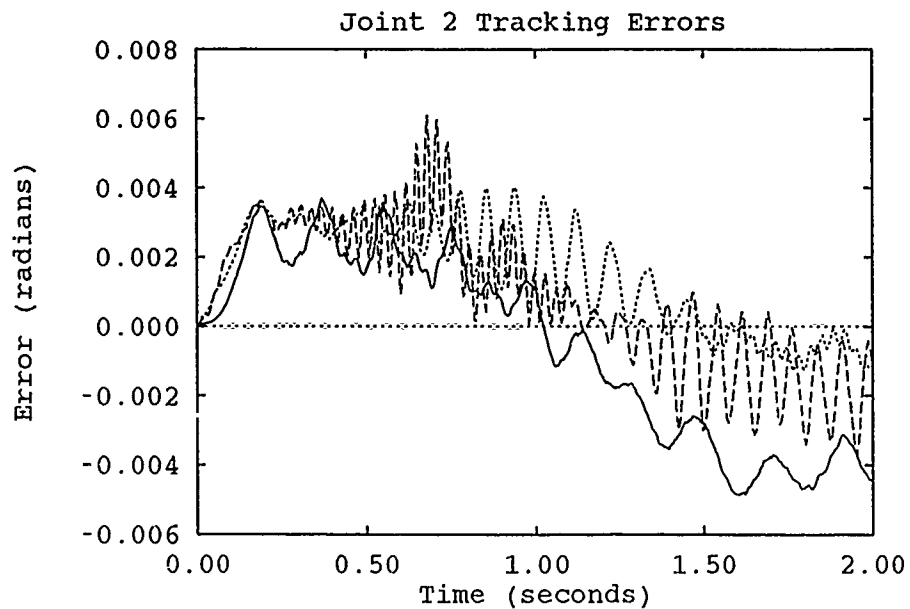


Figure P.2. The Effects of Trajectory 0 Tuning Parameters on Trajectories 2 and 3

—	Trajectory 0	Trajectory 3
- - -	Trajectory 2		

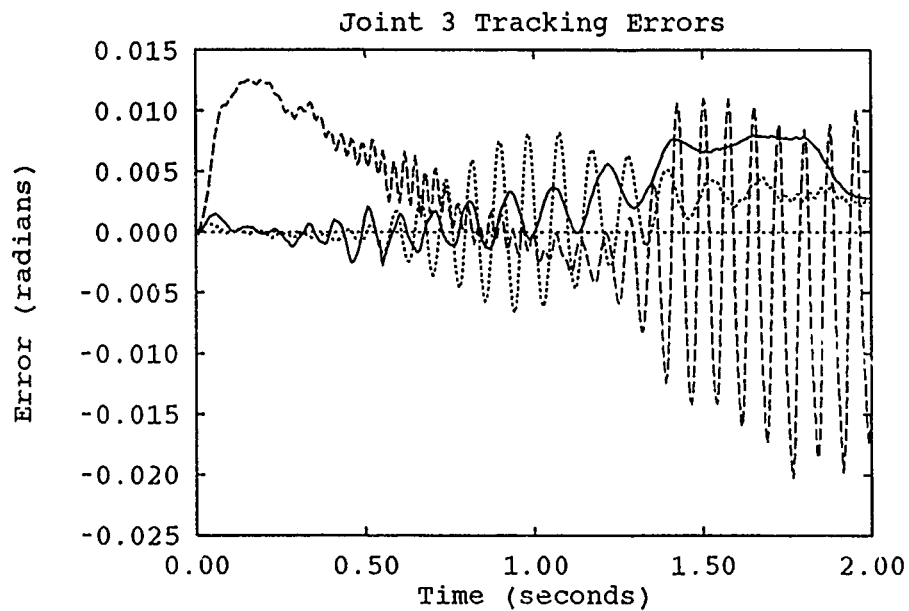


Figure P.3. The Effects of Trajectory 0 Tuning Parameters on Trajectories 2 and 3

—	Trajectory 0	Trajectory 3
- - -	Trajectory 2		

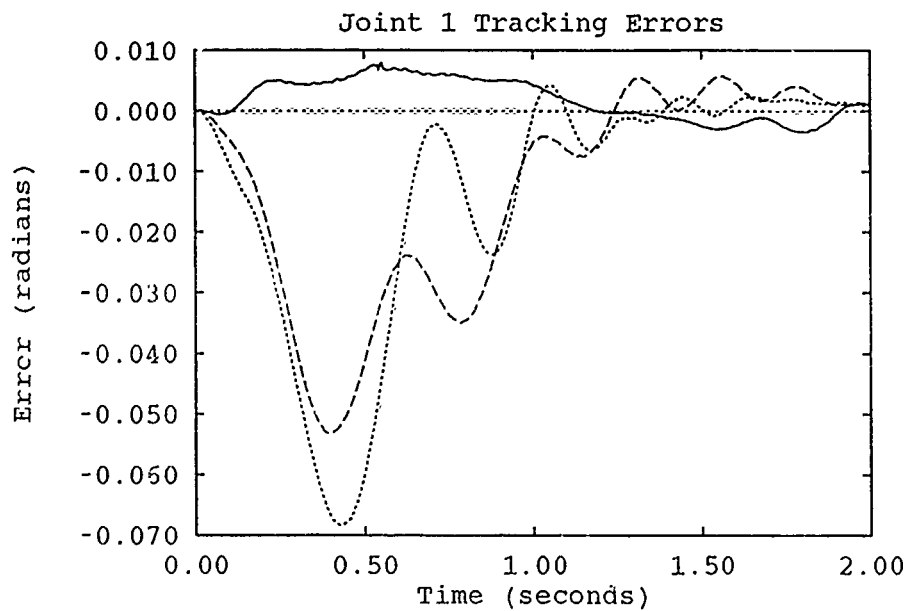


Figure P.4. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5

—	Trajectory 0	Trajectory 5
- - -	Trajectory 4		

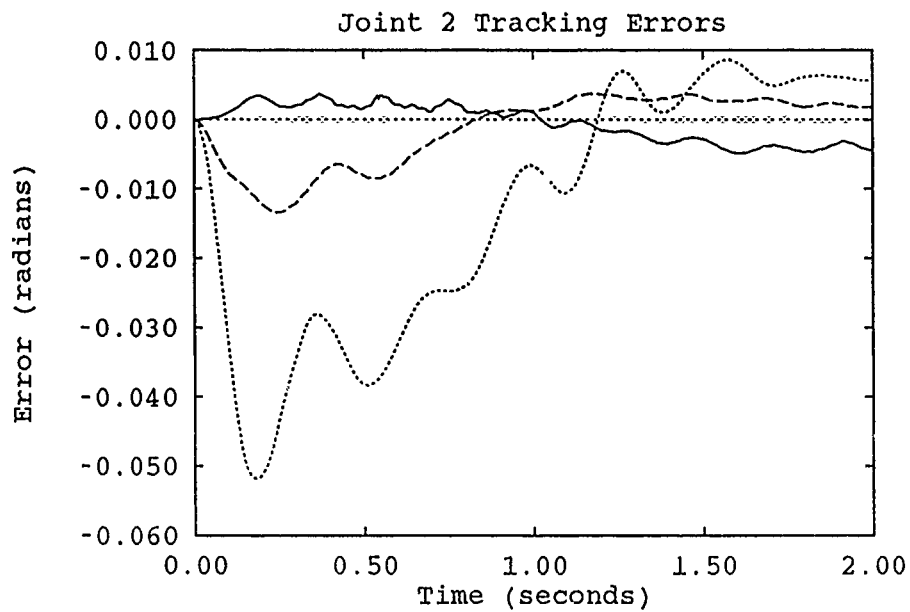


Figure P.5. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5

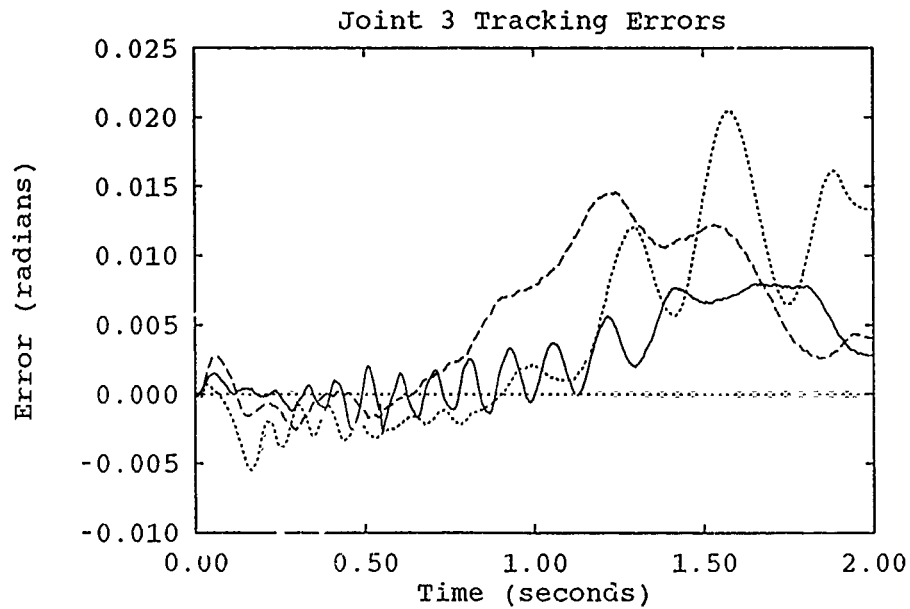


Figure P.6. The Effects of Trajectory 0 Tuning Parameters on Trajectories 4 and 5

—	Trajectory 0	Trajectory 5
- - -	Trajectory 4		

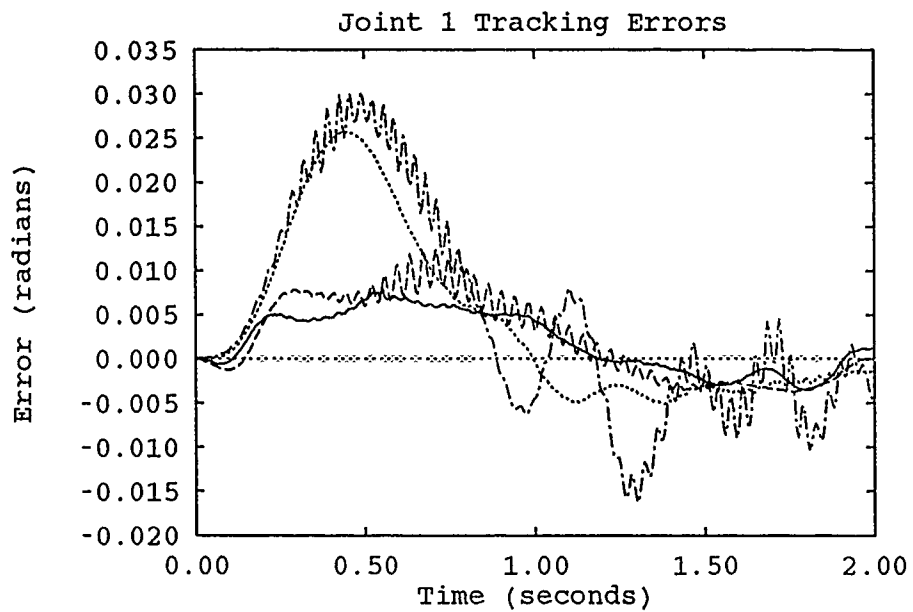


Figure P.7. Payload Effects on Trajectory 0 Tuning Parameters

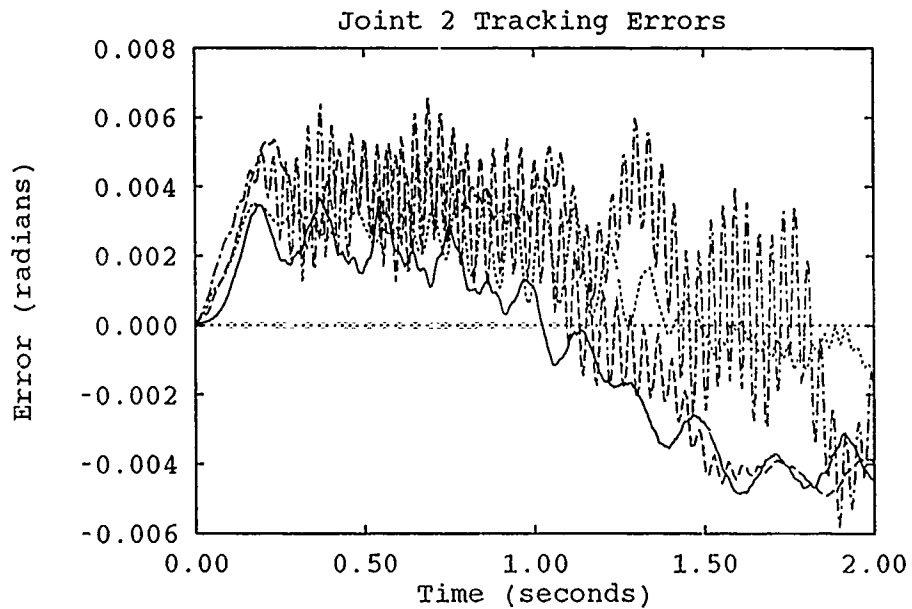


Figure P.8. Payload Effects on Trajectory 0 Tuning Parameters

————	Trajectory 0 - No Load	Trajectory 3 - No Load
-----	Trajectory 0 - 2kg Load	-.-.-.-	Trajectory 3 - 2kg Load

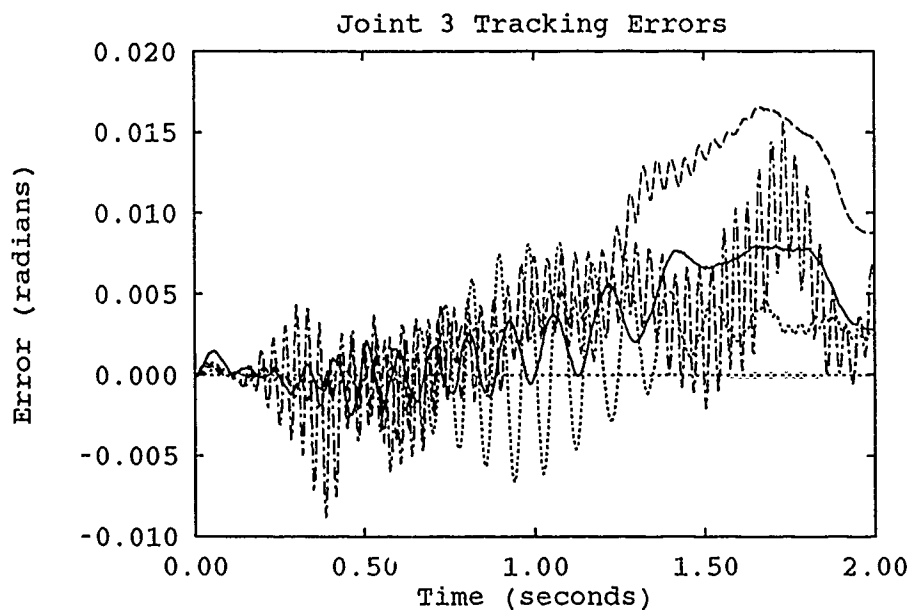


Figure P.9. Payload Effects on Trajectory 0 Tuning Parameters

—	Trajectory 0 - No Load	Trajectory 3 - No Load
----	Trajectory 0 - 2kg Load	- · - · -	Trajectory 3 - 2kg Load

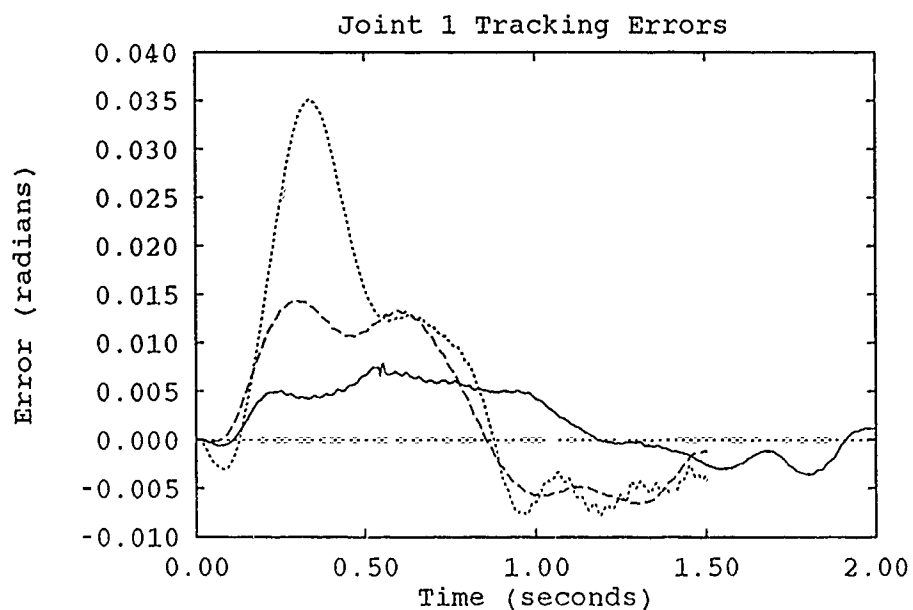


Figure P.10. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6

—	Trajectory 0	Trajectory 6
----	Trajectory 1		

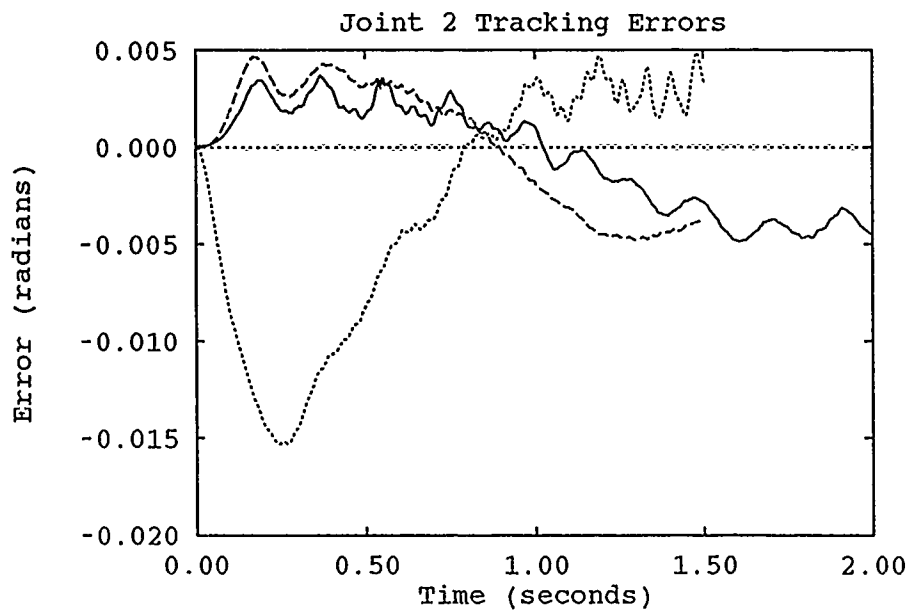


Figure P.11. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6

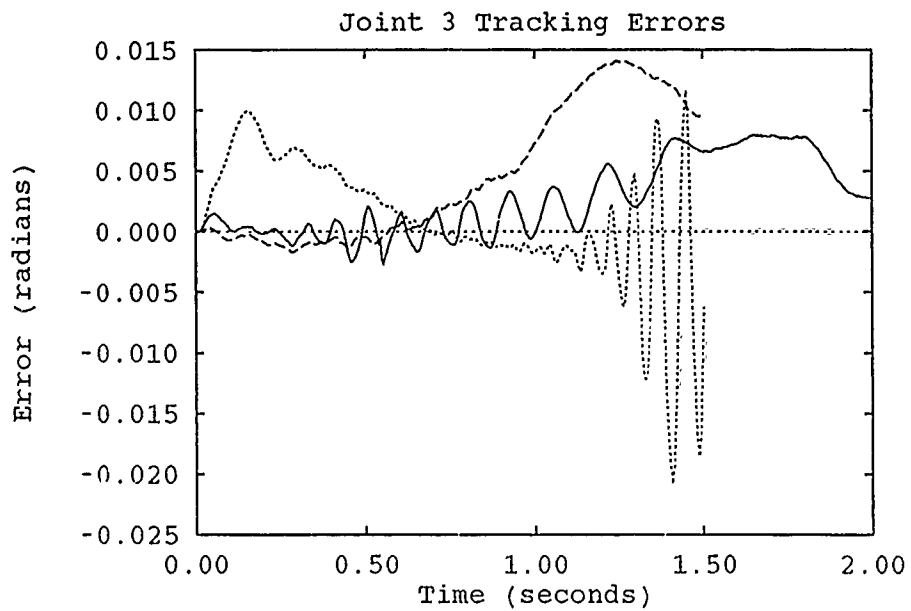


Figure P.12. The Effects of Trajectory 0 Tuning Parameters on Trajectories 1 and 6

—	Trajectory 0	Trajectory 6
- - - -	Trajectory 1		

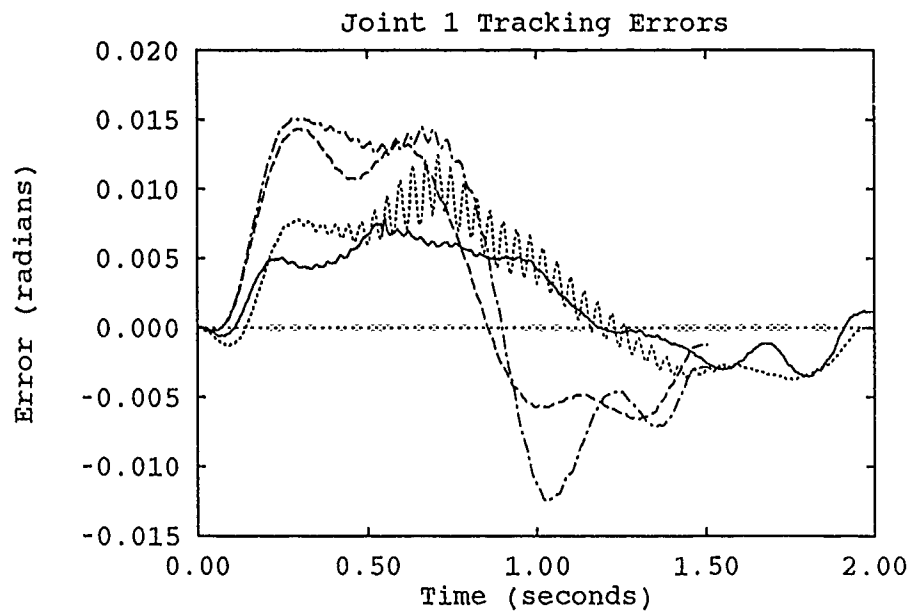


Figure P.13. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec

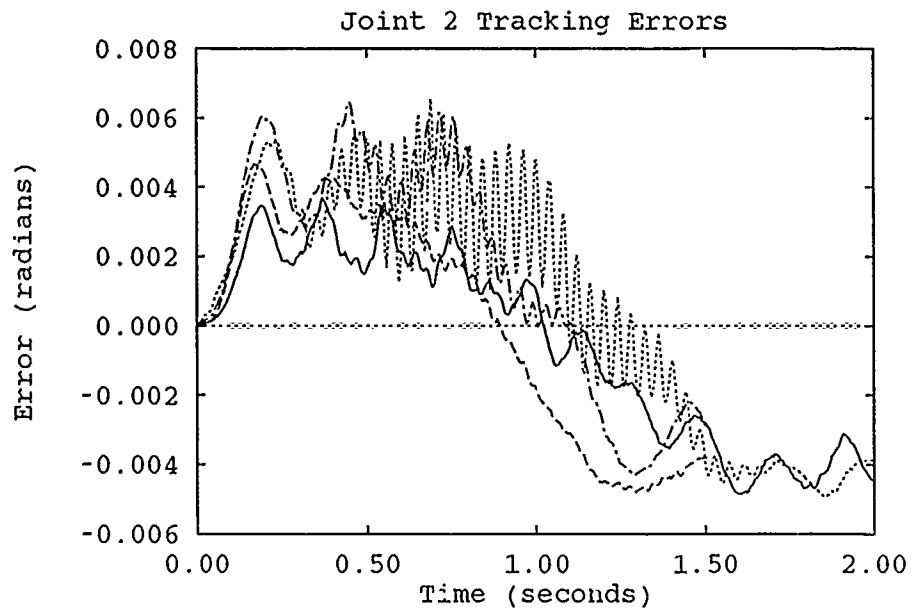


Figure P.14. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec

—	Trajectory 0	Trajectory 0 with payload
----	Trajectory 1	-.-.-.-	Trajectory 1 with payload

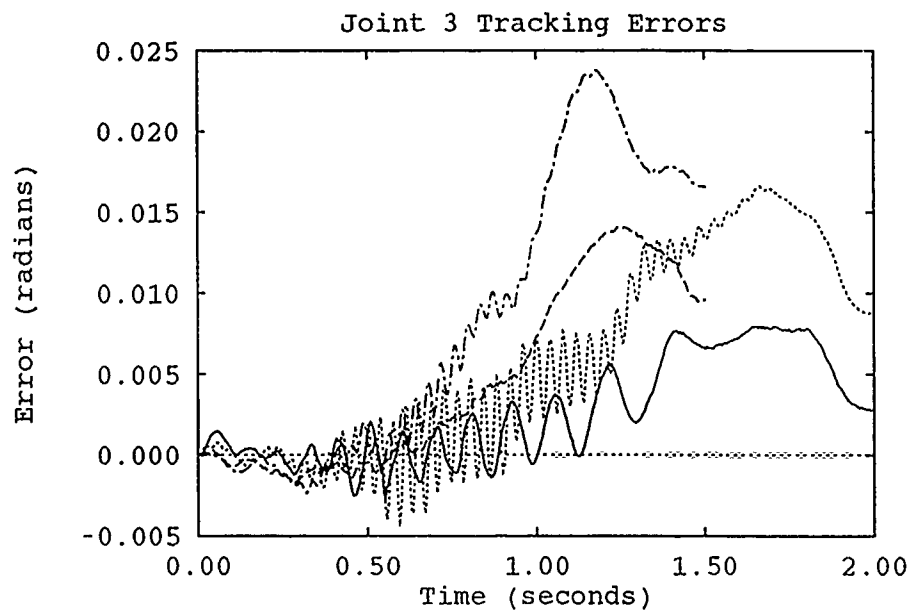


Figure P.15. The effects on Trajectory 0 Tuning Parameters on Speed - 2 sec vs 1.5 sec

—	Trajectory 0	Trajectory 0 with payload
----	Trajectory 1	-.-.-.-	Trajectory 1 with payload

Appendix Q. *Comparisons of Tarokh and Seraji Control Algorithms*

This appendix contains comparison plots of Tarokh's and Seraji's adaptive control algorithms for all test trajectories.

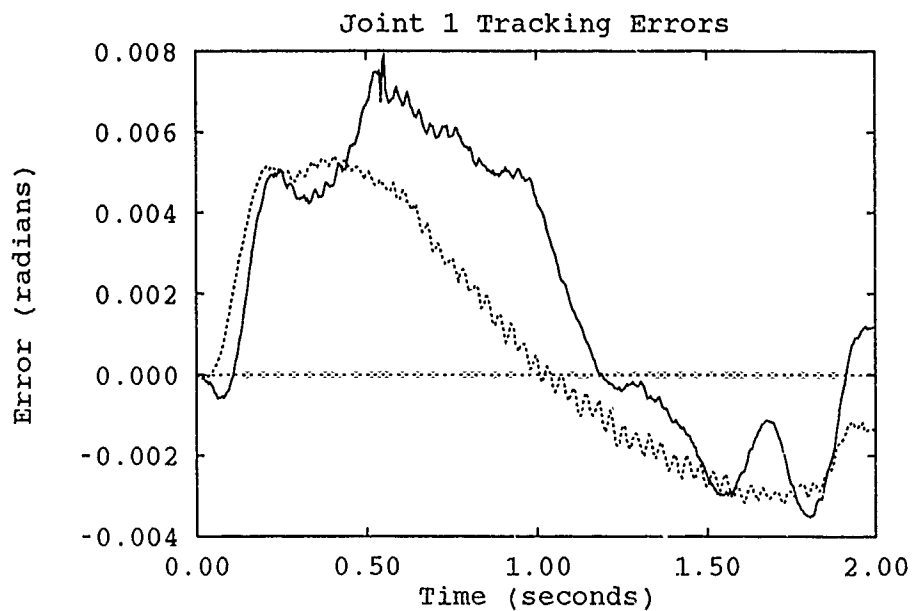


Figure Q.1. Comparison of Tarokh and Seraji Controllers - Trajectory 0

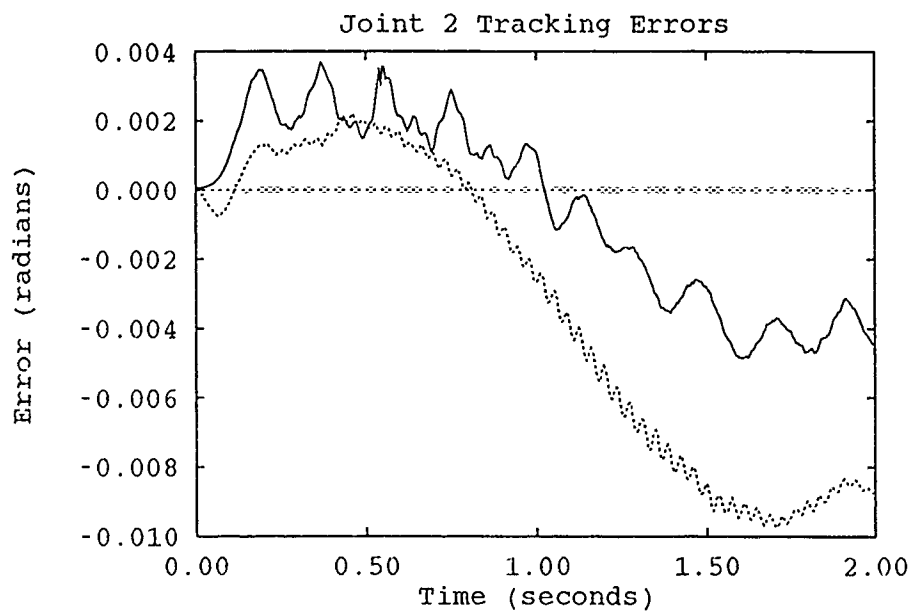
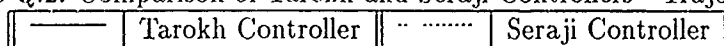


Figure Q.2. Comparison of Tarokh and Seraji Controllers - Trajectory 0



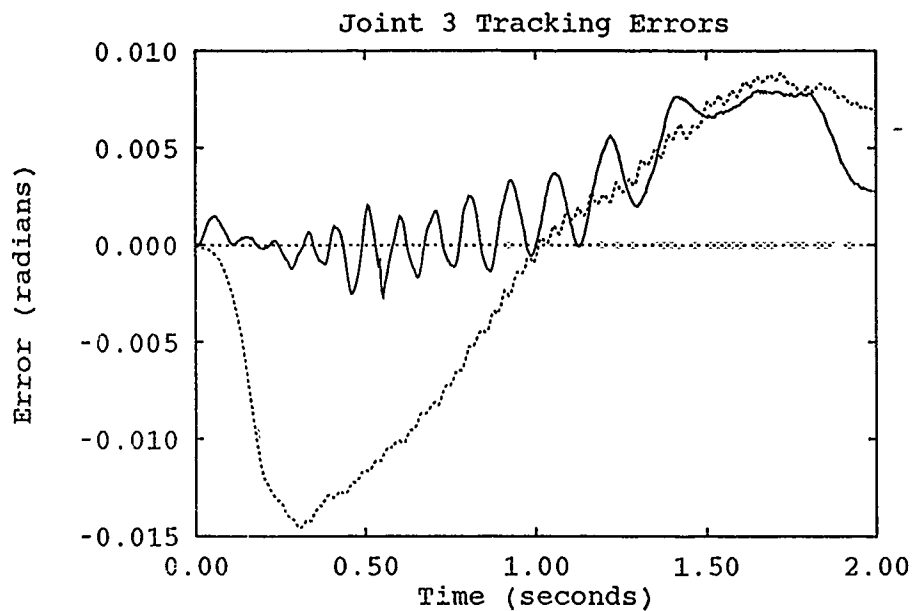


Figure Q.3. Comparison of Tarokh and Seraji Controllers - Trajectory 0

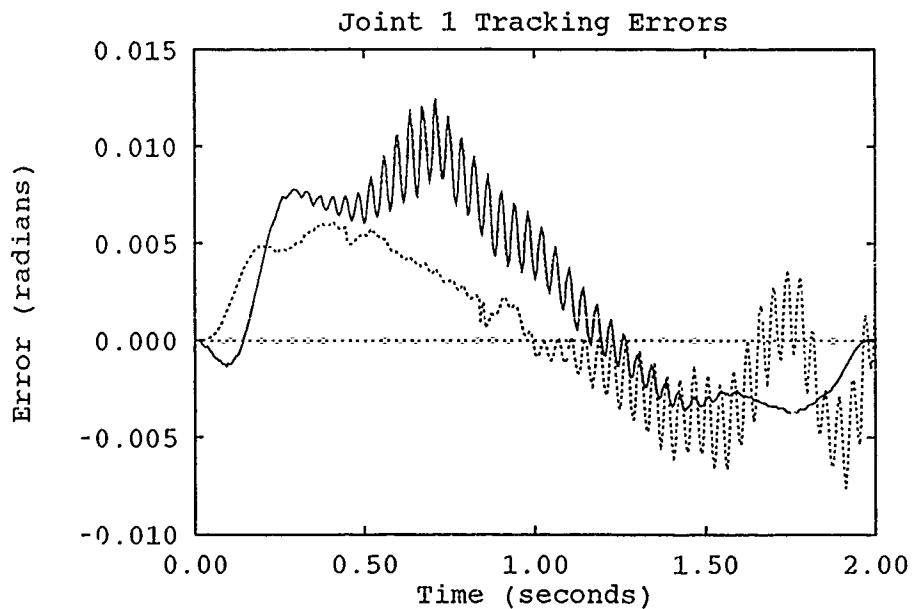
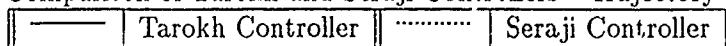


Figure Q.4. Comparison of Tarokh and Seraji Controllers - Trajectory 0 w/ payload



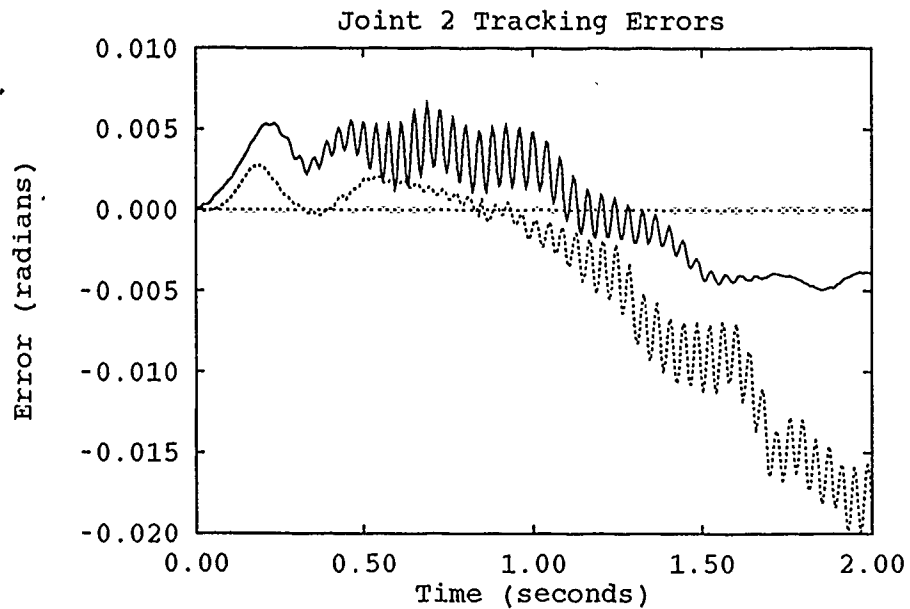


Figure Q.5. Comparison of Tarokh and Seraji Controllers - Trajectory 0 w/ Payload

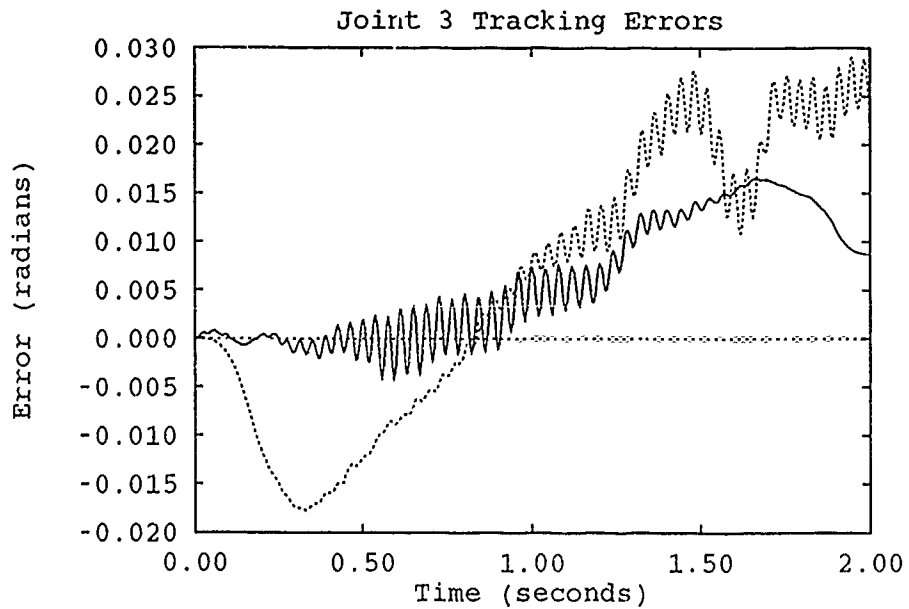
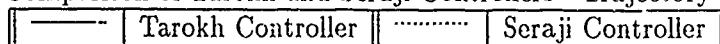


Figure Q.6. Comparison of Tarokh and Seraji Controllers - Trajectory 0 w/ Payload



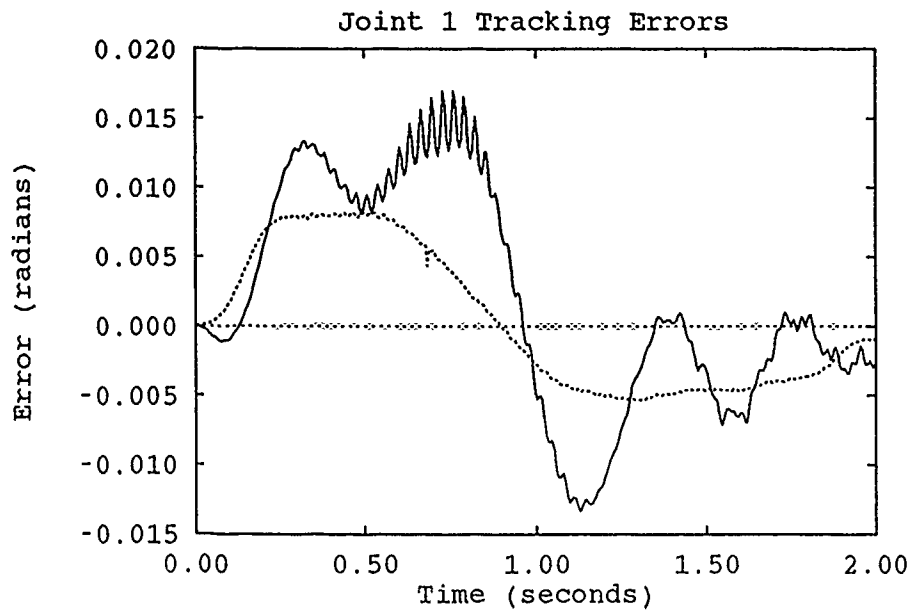


Figure Q.7. Comparison of Tarokh and Seraji Controllers - Trajectory 2

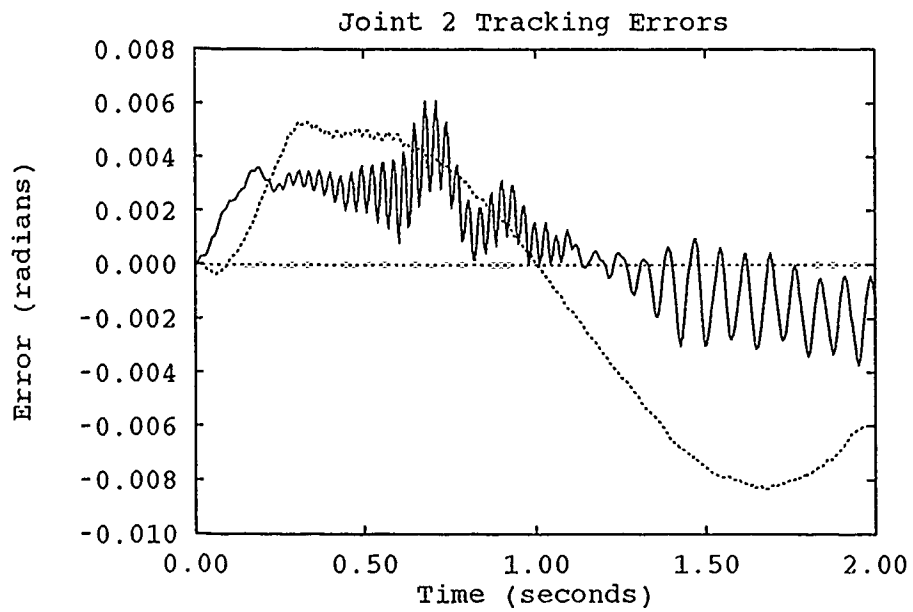
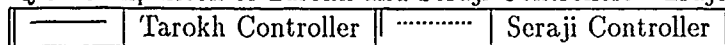


Figure Q.8. Comparison of Tarokh and Seraji Controllers - Trajectory 2



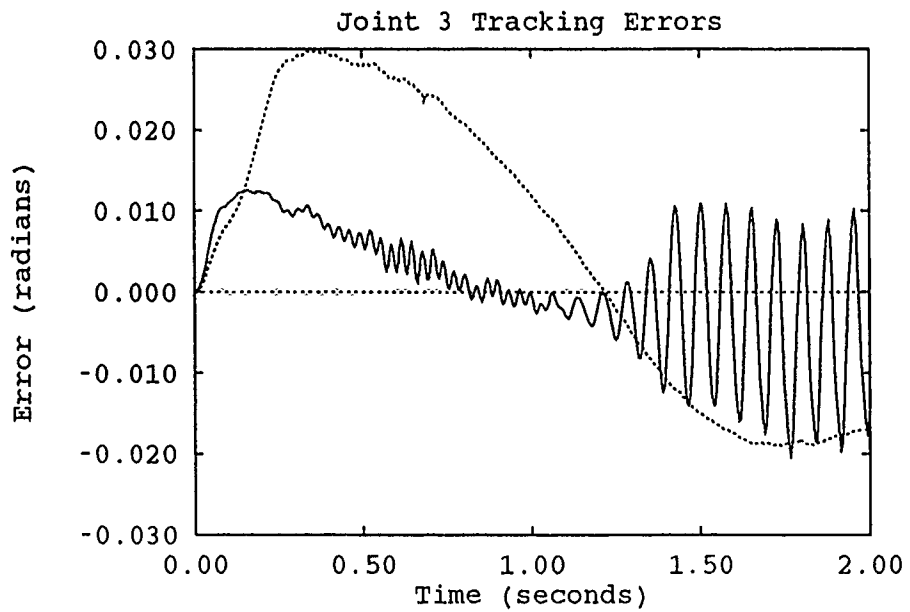


Figure Q.9. Comparison of Tarokh and Seraji Controllers - Trajectory 2

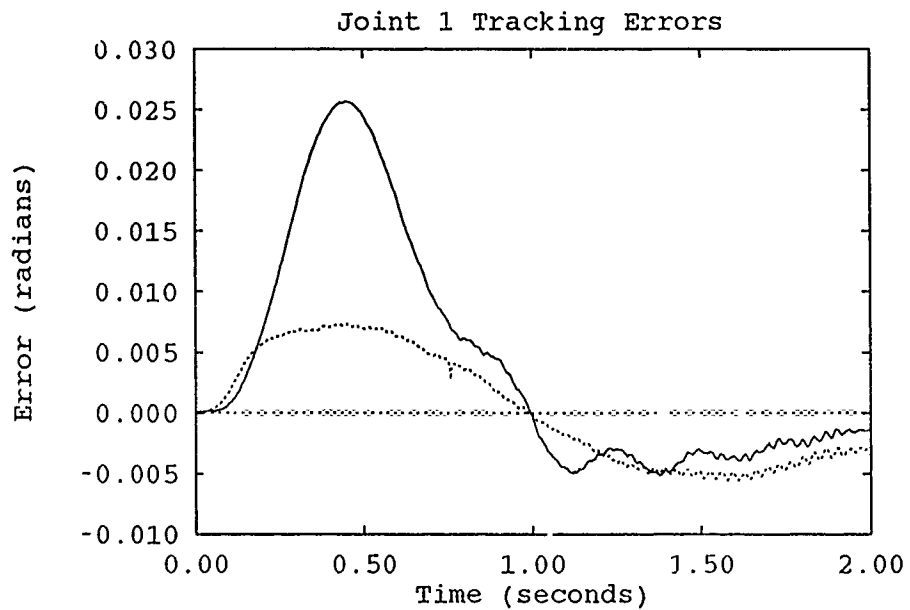
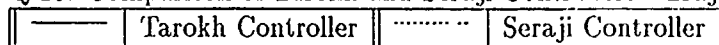


Figure Q.10. Comparison of Tarokh and Seraji Controllers - Trajectory 3



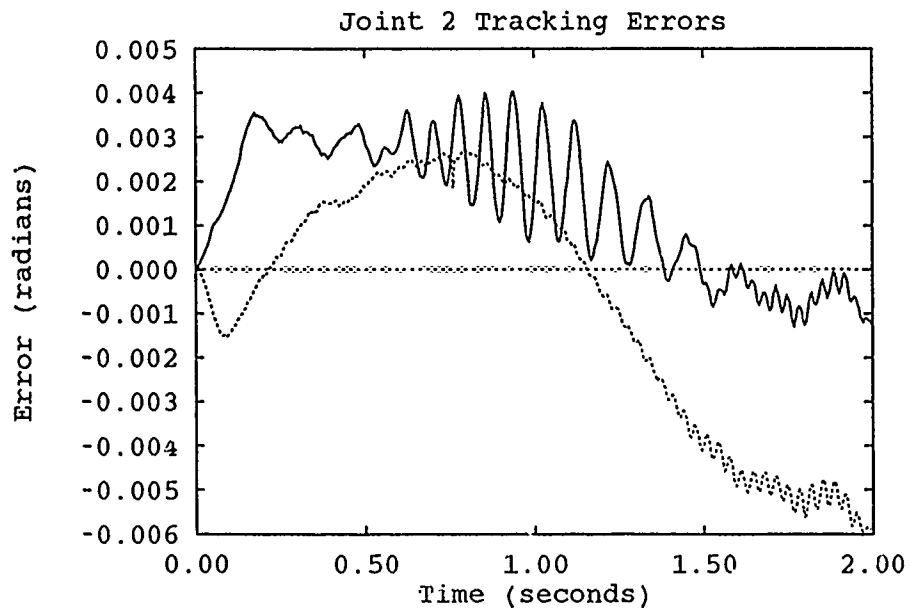


Figure Q.11. Comparison of Tarokh and Seraji Controllers - Trajectory 3

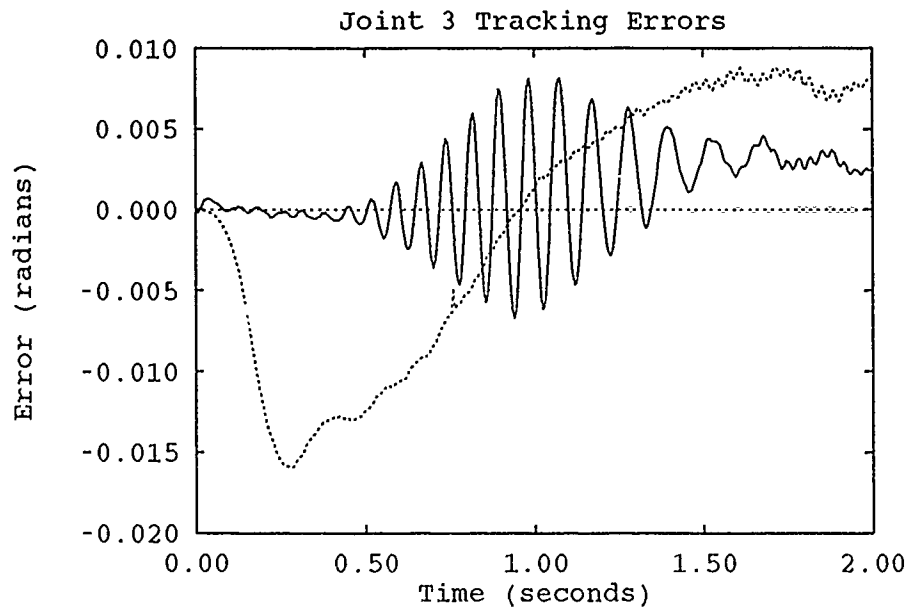
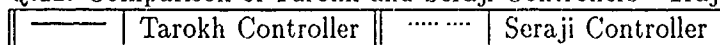


Figure Q.12. Comparison of Tarokh and Seraji Controllers - Trajectory 3



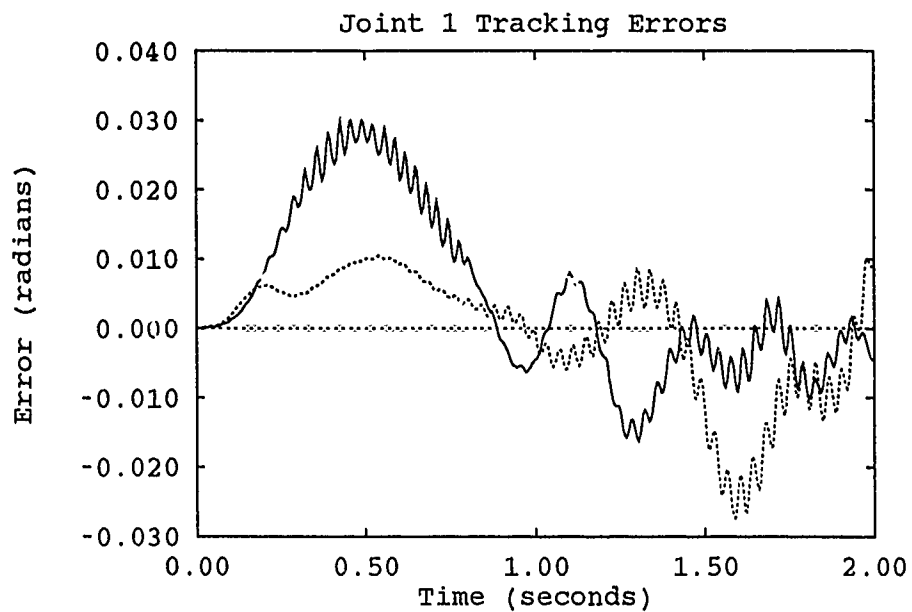


Figure Q.13. Comparison of Tarokh and Seraji Controllers - Trajectory 3 w/ Payload

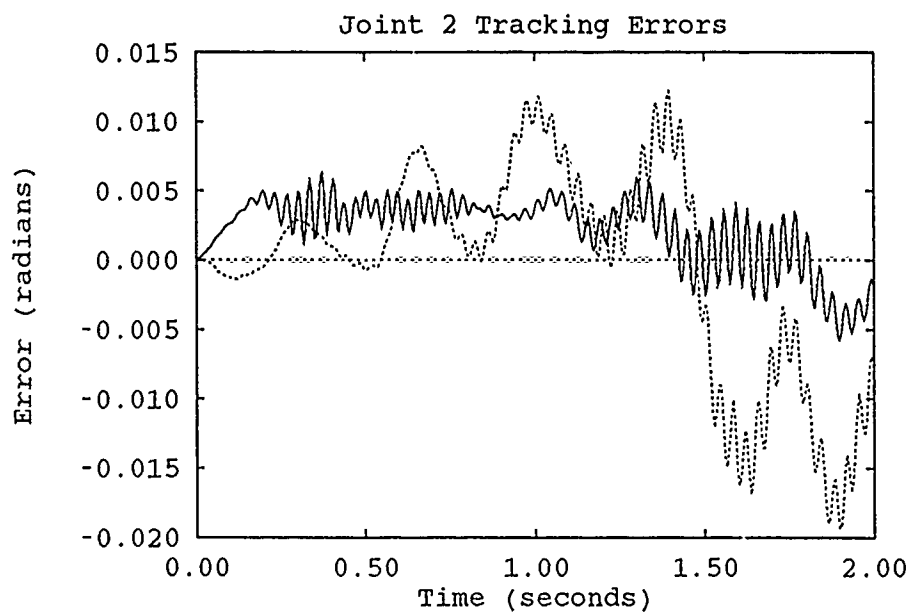
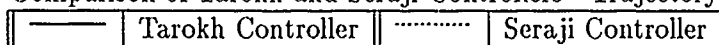


Figure Q.14. Comparison of Tarokh and Seraji Controllers - Trajectory 3 w/ Payload



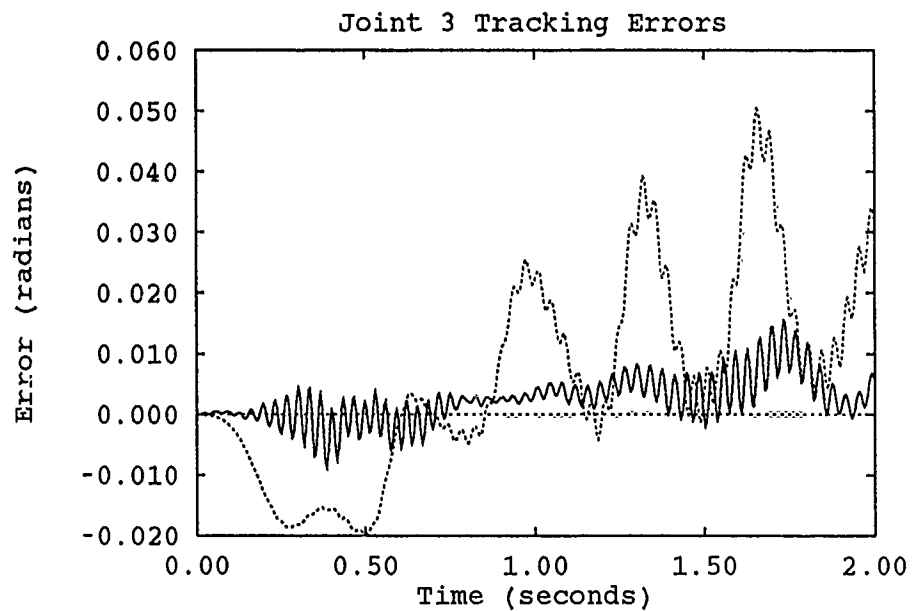


Figure Q.15. Comparison of Tarokh and Seraji Controllers - Trajectory 3 w/ Payload

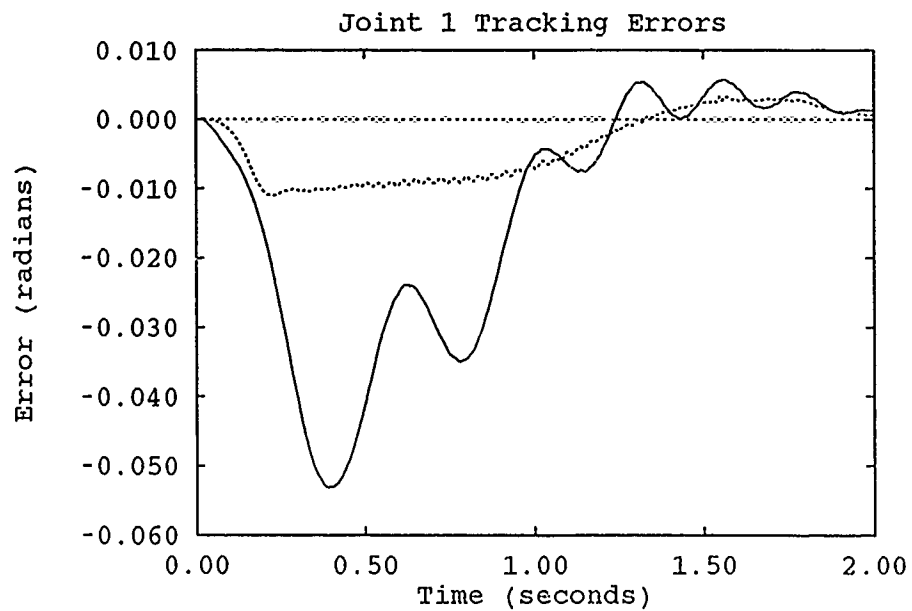
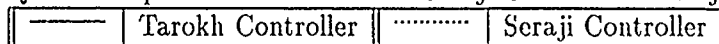


Figure Q.16. Comparison of Tarokh and Seraji Controllers - Trajectory 4



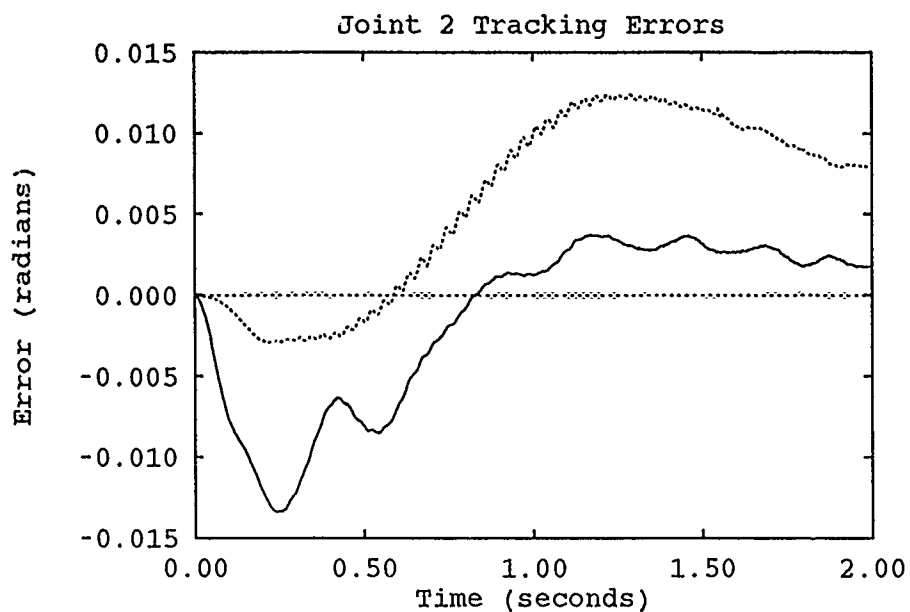


Figure Q.17. Comparison of Tarokh and Seraji Controllers - Trajectory 4

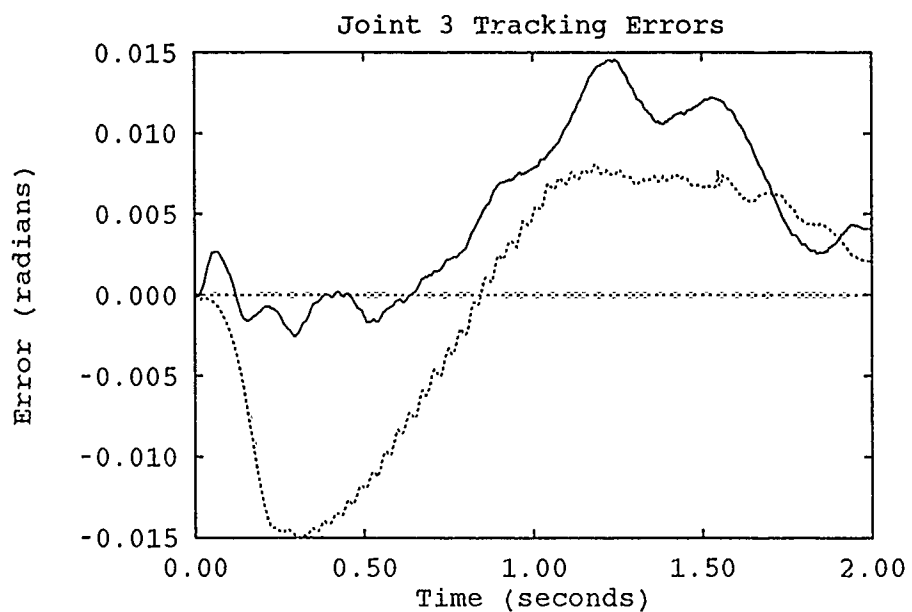
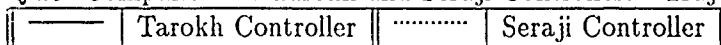


Figure Q.18. Comparison of Tarokh and Seraji Controllers - Trajectory 4



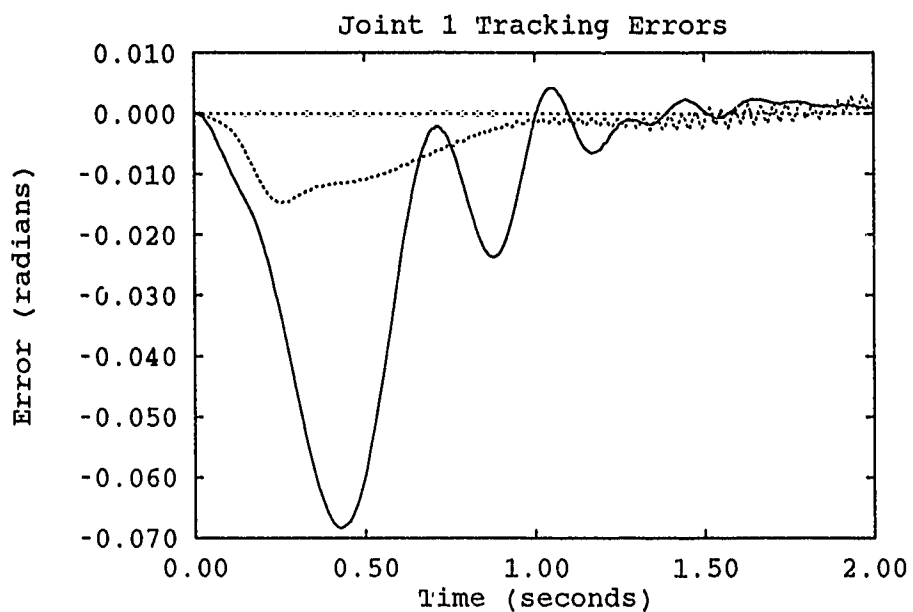


Figure Q.19. Comparison of Tarokh and Seraji Controllers - Trajectory 5

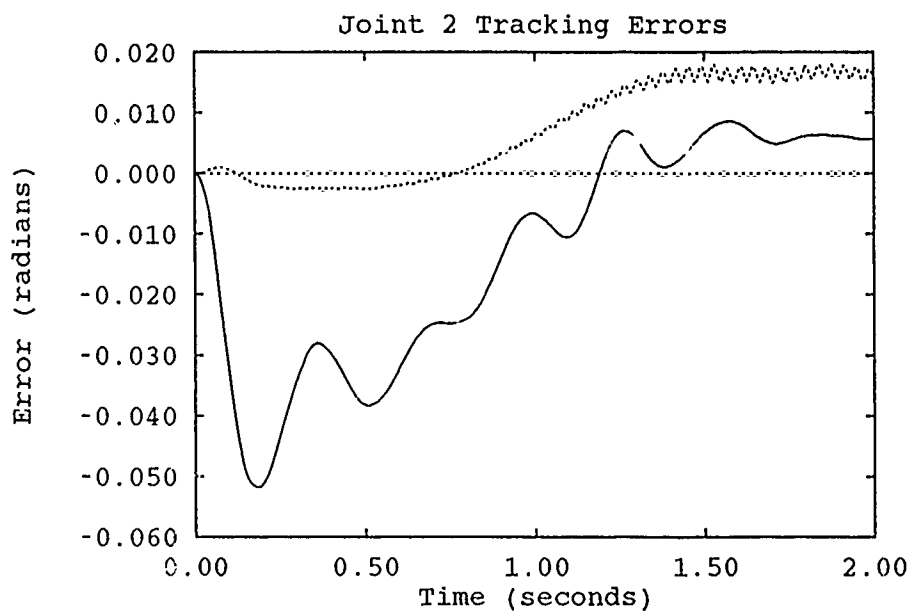
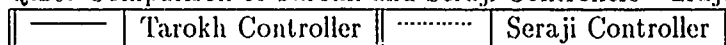


Figure Q.20. Comparison of Tarokh and Seraji Controllers - Trajectory 5



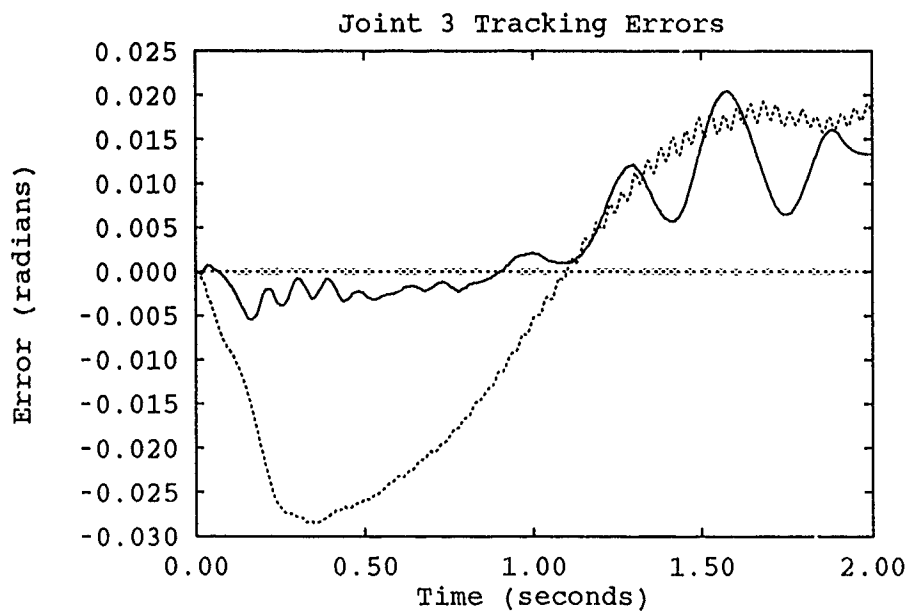


Figure Q.21. Comparison of Tarokh and Seraji Controllers - Trajectory 5

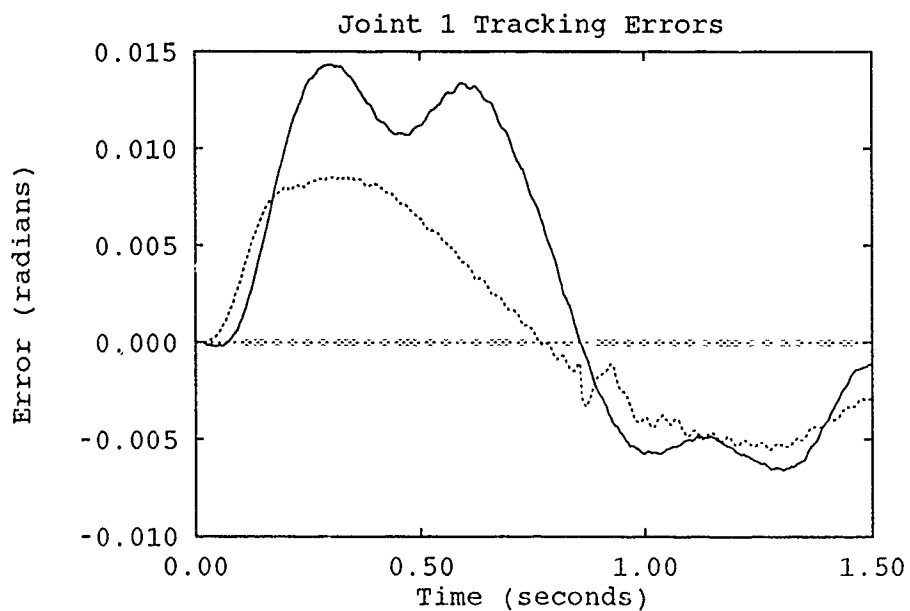
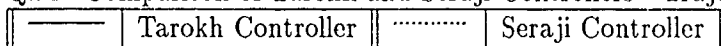


Figure Q.22. Comparison of Tarokh and Seraji Controllers - Trajectory 1



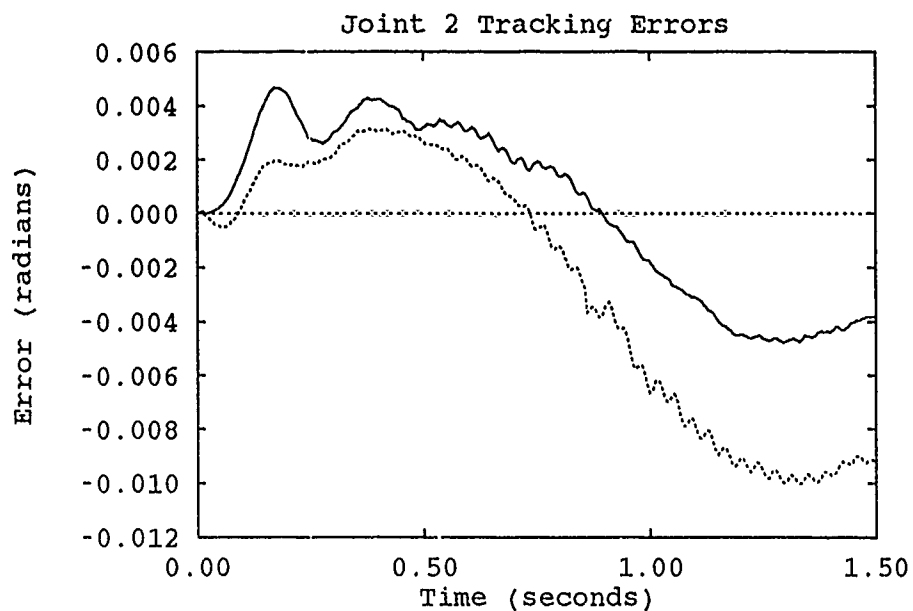


Figure Q.23. Comparison of Tarokh and Seraji Controllers - Trajectory 1

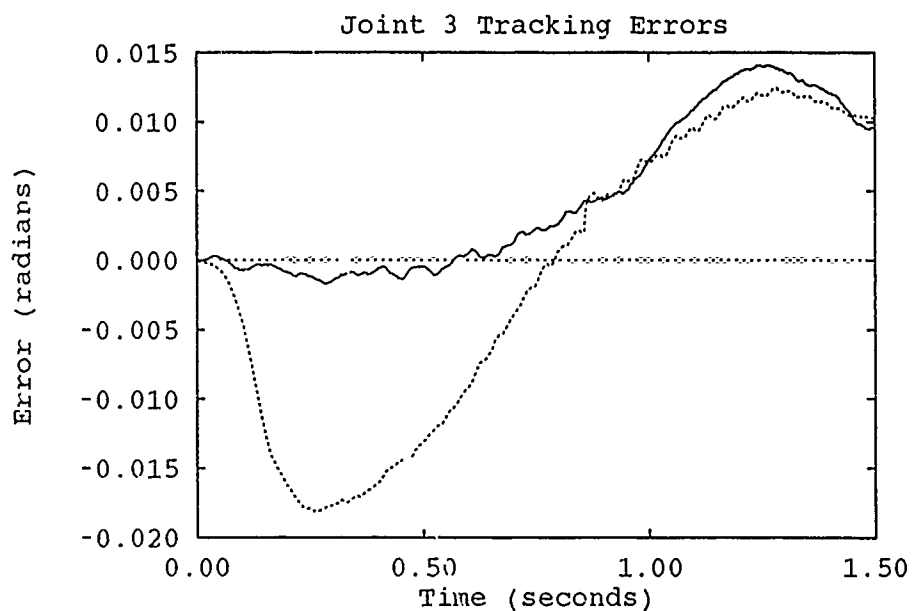
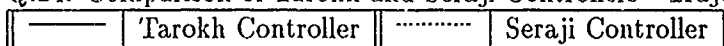


Figure Q.24. Comparison of Tarokh and Seraji Controllers - Trajectory 1



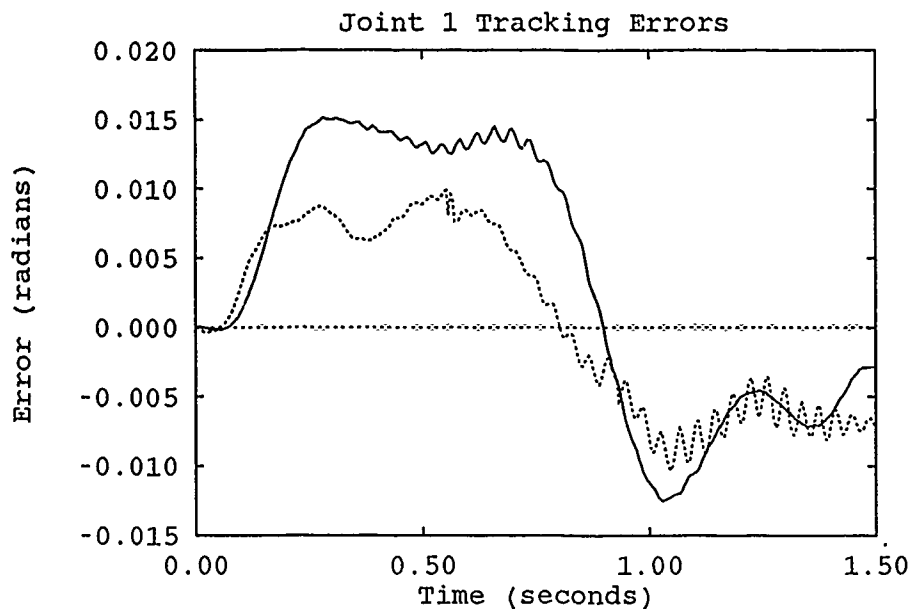


Figure Q.25. Comparison of Tarokh and Seraji Controllers - Trajectory 1 w/ Payload

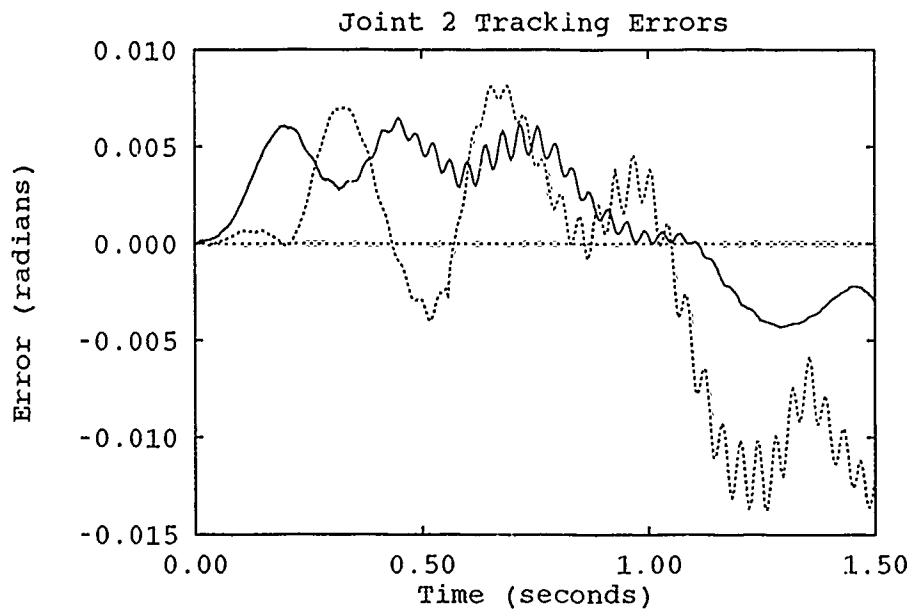
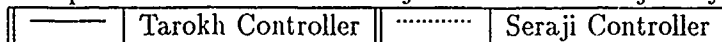


Figure Q.26. Comparison of Tarokh and Seraji Controllers - Trajectory 1 w/ Payload



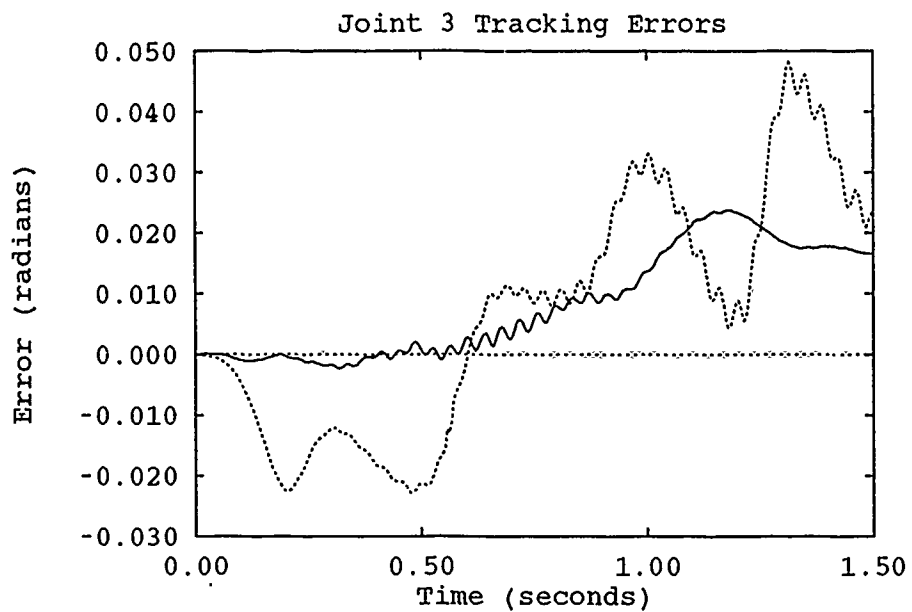


Figure Q.27. Comparison of Tarokh and Seraji Controllers - Trajectory 1 w/ Payload

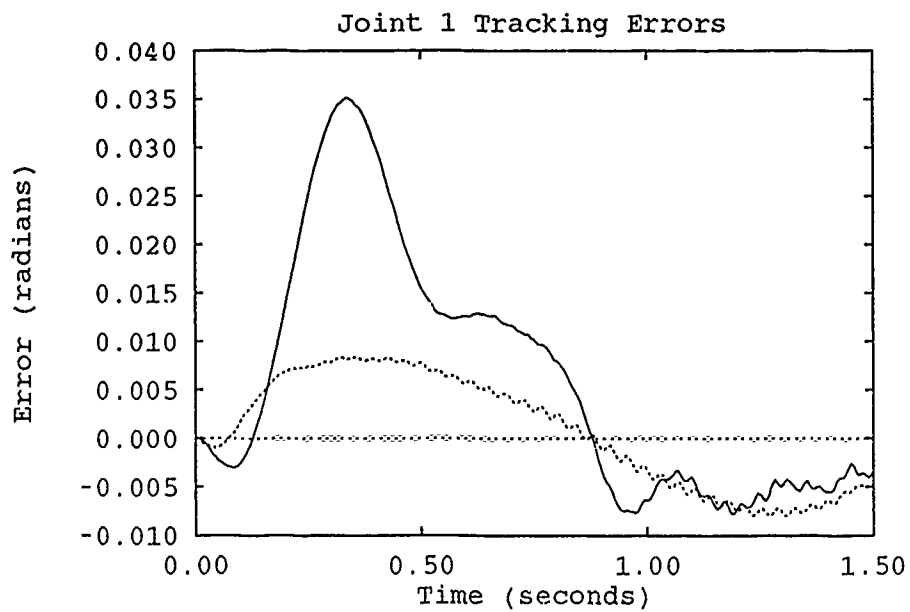
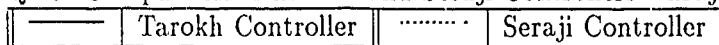


Figure Q.28. Comparison of Tarokh and Seraji Controllers - Trajectory 6



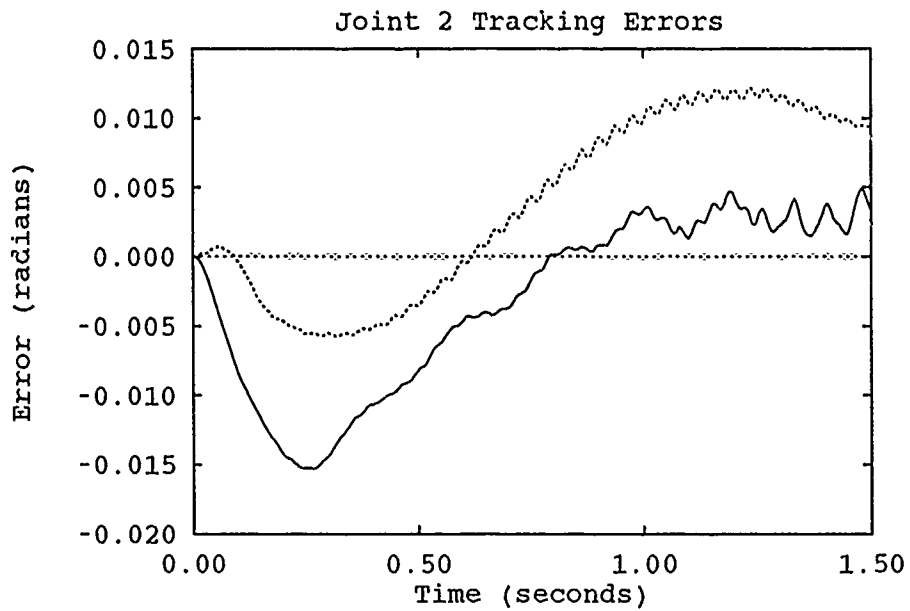


Figure Q.29. Comparison of Tarokh and Seraji Controllers - Trajectory 6

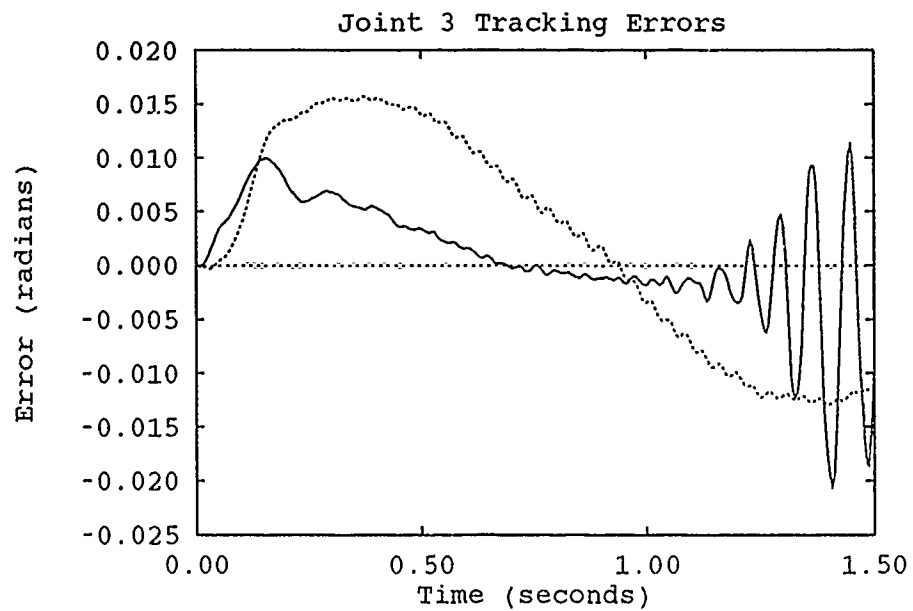
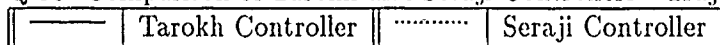


Figure Q.30. Comparison of Tarokh and Seraji Controllers - Trajectory 6



Appendix R. *Comparisons of MBAIC and SMBC Controllers*

This appendix contains comparison plots of the tracking errors generated by the Model-Based Auxiliary Input Controllers (MBAIC) constructed from the auxiliary torque components of Tarokh's and Seraji's adaptive control algorithms. These plots illustrate the improvements the MBAIC controllers have over the SMBC controller composed of only PD feedback torques and a model-based feedforward torque.

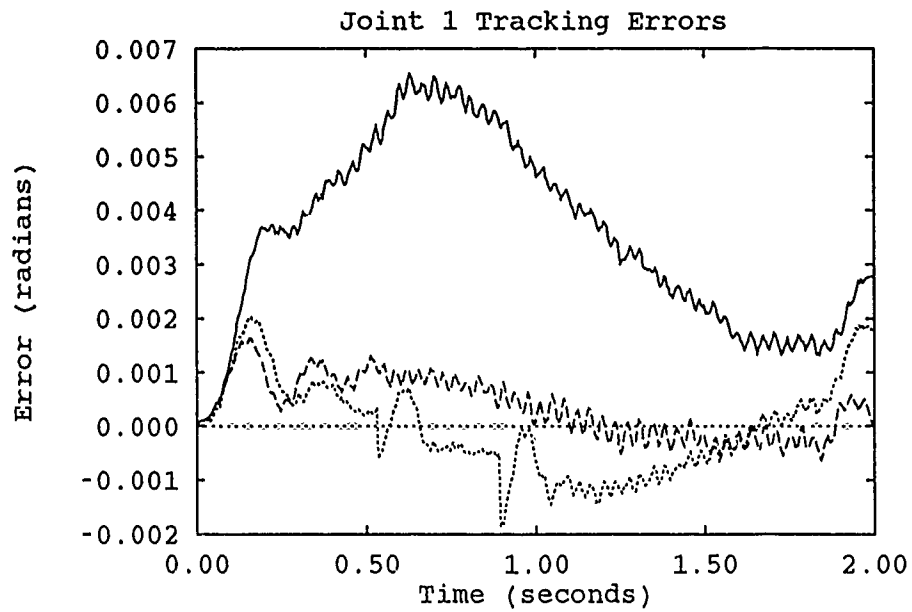


Figure R.1. Comparison of MBAIC and SMBC Controllers - Trajectory 0

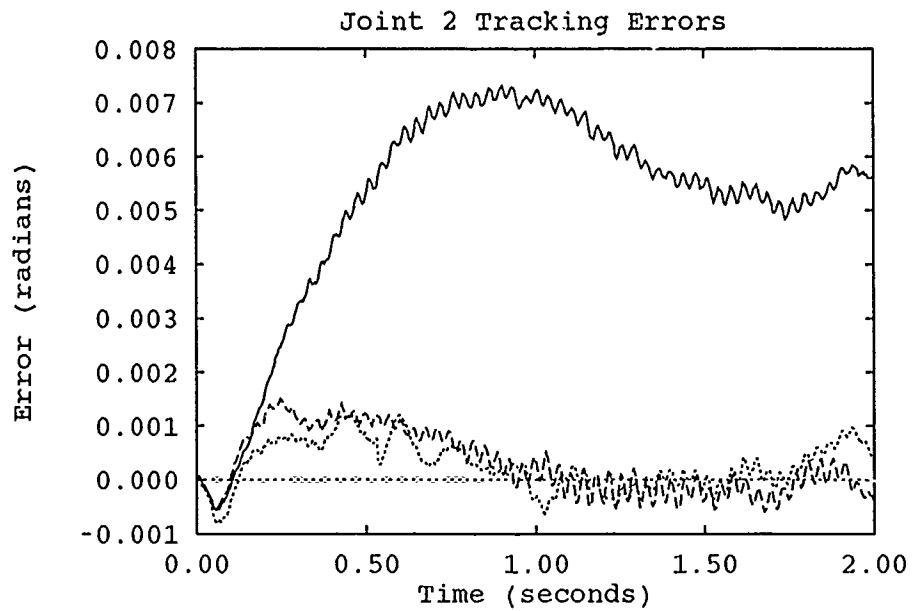


Figure R.2. Comparison of MBAIC and SMBC Controllers - Trajectory 0

----	Tarokh-Based MBAIC	—	SMBC Controller
.....	Seraji-Based MBAIC		

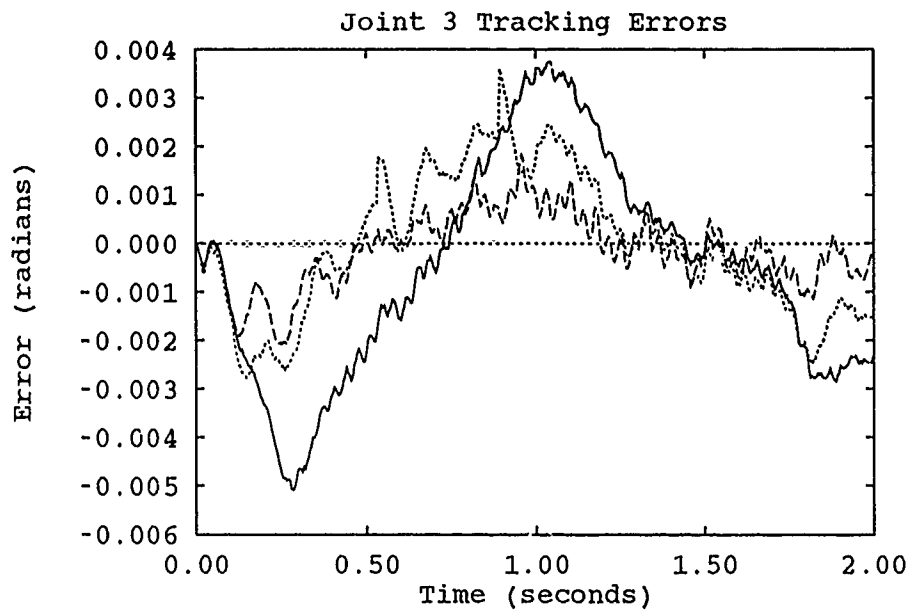


Figure R.3. Comparison of MBAIC and SMBC Controllers - Trajectory 0

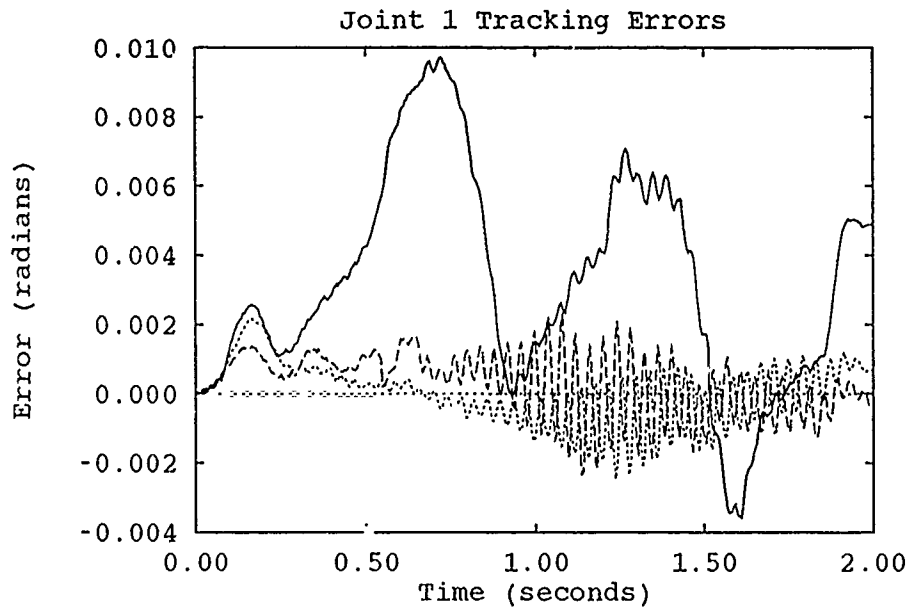


Figure R.4. Comparison of MBAIC and SMBC Controllers - Trajectory 0 w/ payload

-----	Tarokh-Based MBAIC	————	SMBC Controller
.....	Seraji-Based MBAIC		

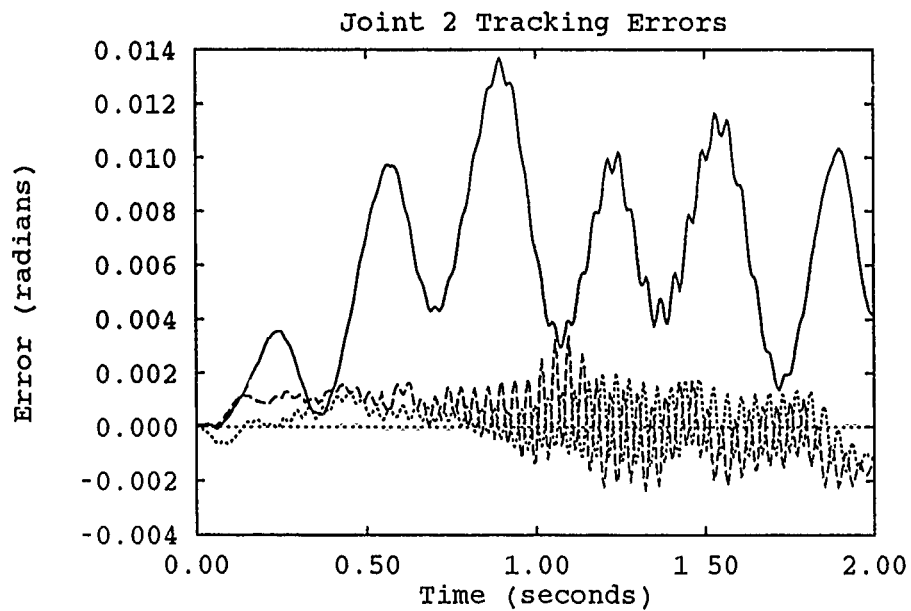


Figure R.5. Comparison of MBAIC and SMBC Controllers - Trajectory 0 w/ Payload

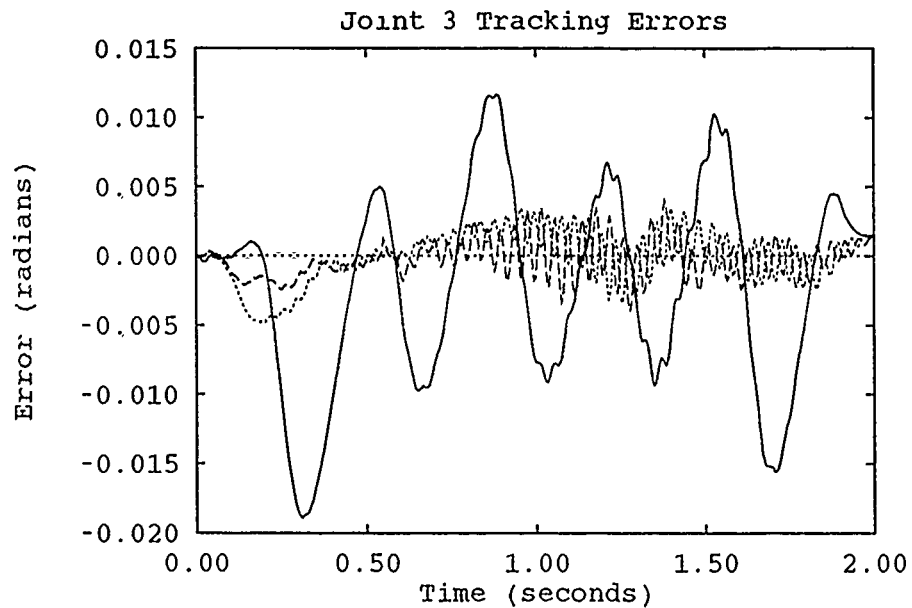


Figure R.6. Comparison of MBAIC and SMBC Controllers - Trajectory 0 w/ Payload

-----	Tarokh-Based MBAIC	————	SMBC Controller
.....	Seraji-Based MBAIC		

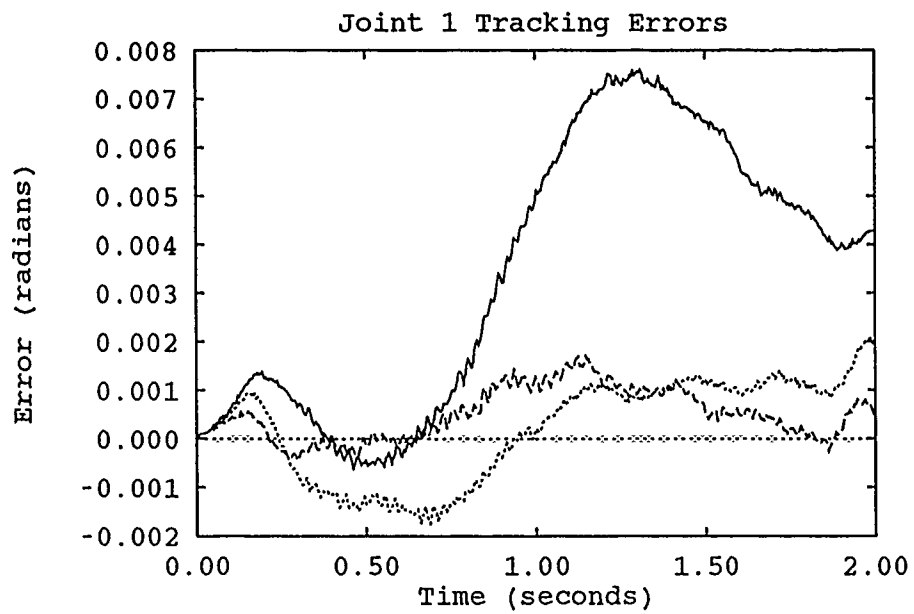


Figure R.7. Comparison of MBAIC and SMBC Controllers - Trajectory 2

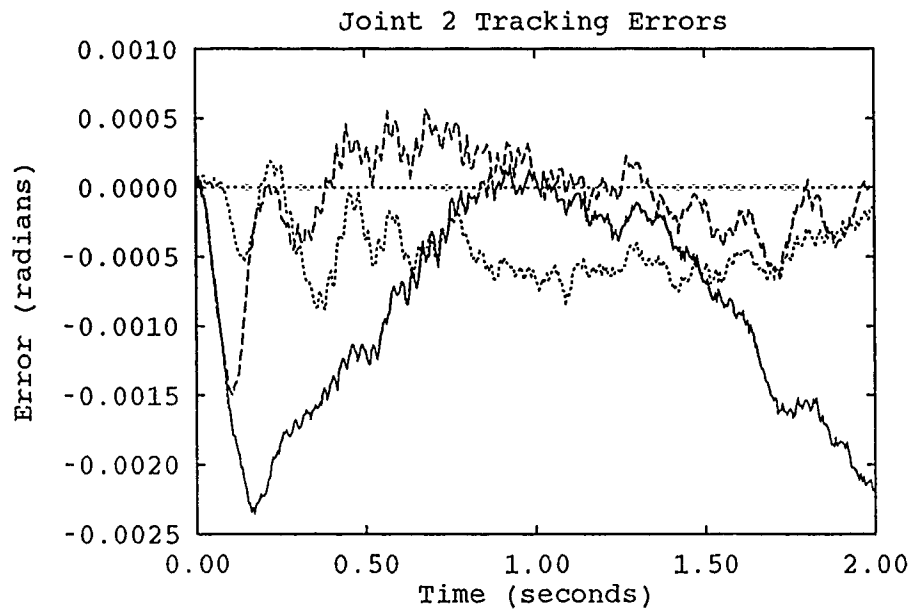


Figure R.8. Comparison of MBAIC and SMBC Controllers - Trajectory 2

----	Tarokh-Based MBAIC	—	SMBC Controller
.....	Seraji-Based MBAIC		

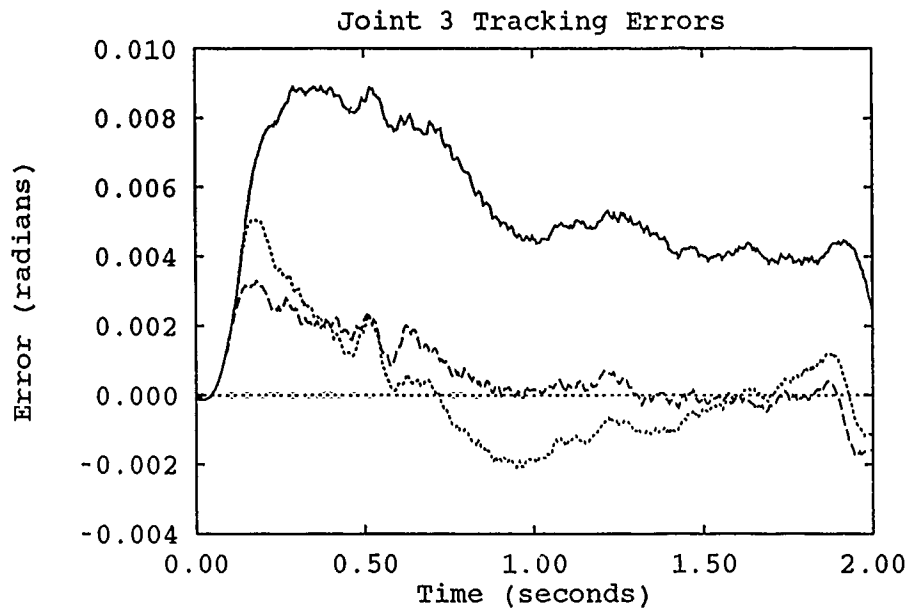


Figure R.9. Comparison of MBAIC and SMBC Controllers - Trajectory 2

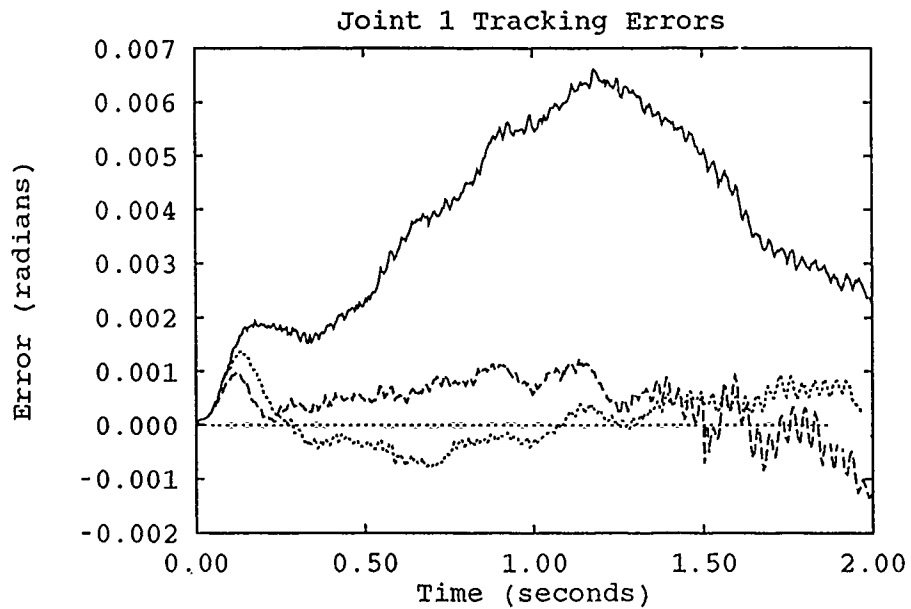


Figure R.10. Comparison of MBAIC and SMBC Controllers - Trajectory 3

----	Tarokh-Based MBAIC	—	SMBC Controller
.....	Seraji-Based MBAIC		

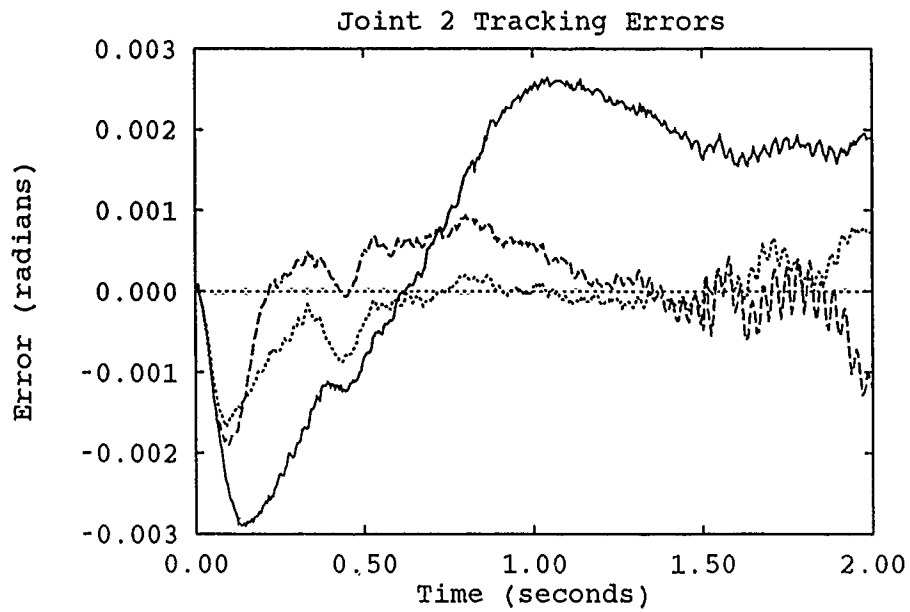


Figure R.11. Comparison of MBAIC and SMBC Controllers - Trajectory 3

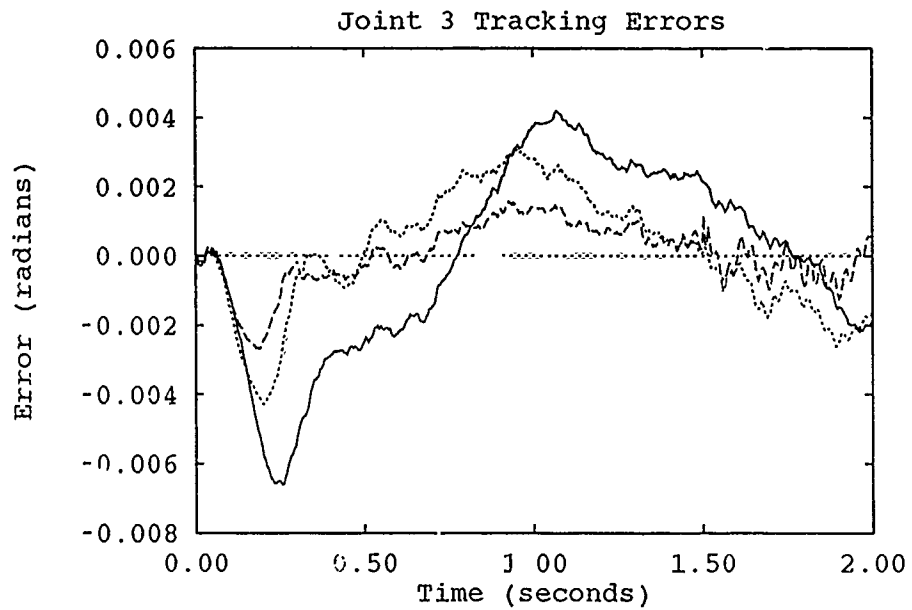


Figure R.12. Comparison of MBAIC and SMBC Controllers - Trajectory 3

----	Tarokh-Based MBAIC	—	SMBC Controller
.....	Seraji-Based MBAIC		

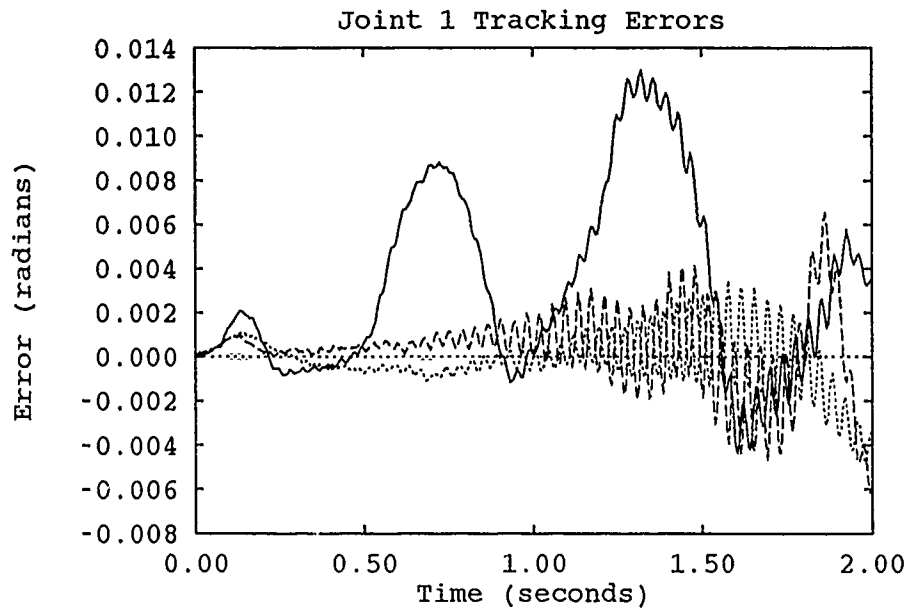


Figure R.13. Comparison of MBAIC and SMBC Controllers - Trajectory 3 w/ Payload

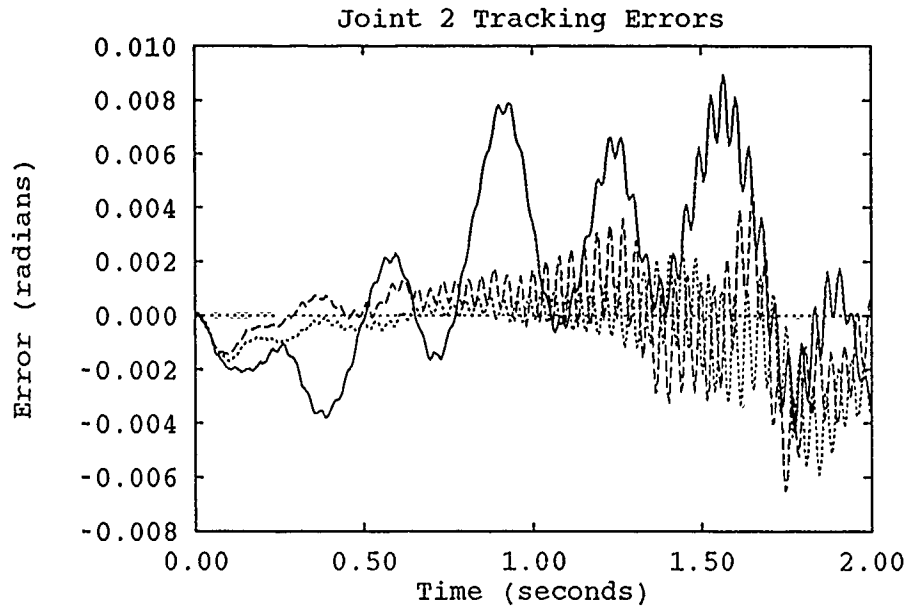


Figure R.14. Comparison of MBAIC and SMBC Controllers - Trajectory 3 w/ Payload

----	Tarokh-Based MBAIC	—	SMBC Controller
.....	Seraji-Based MBAIC		

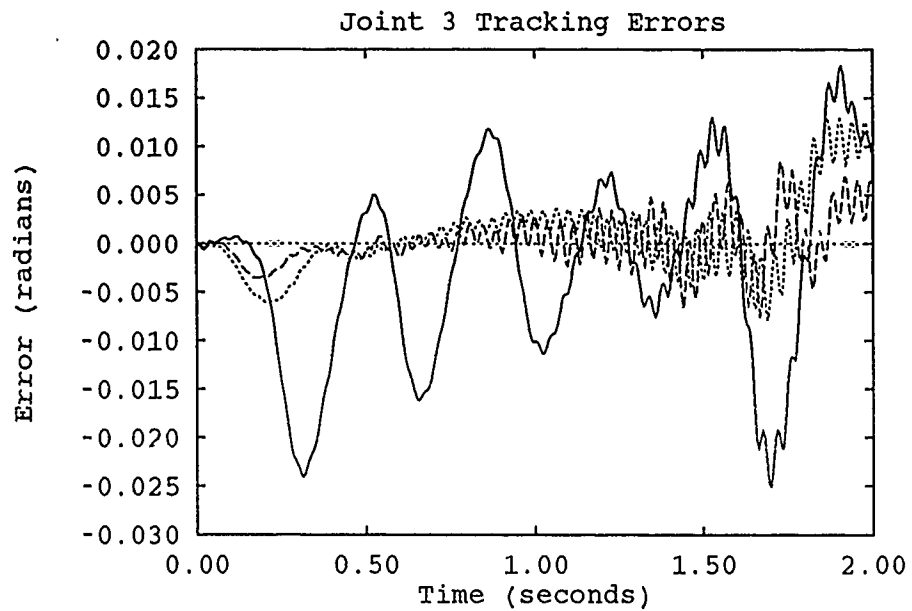


Figure R.15. Comparison of MBAIC and SMBC Controllers - Trajectory 3 w/ Payload

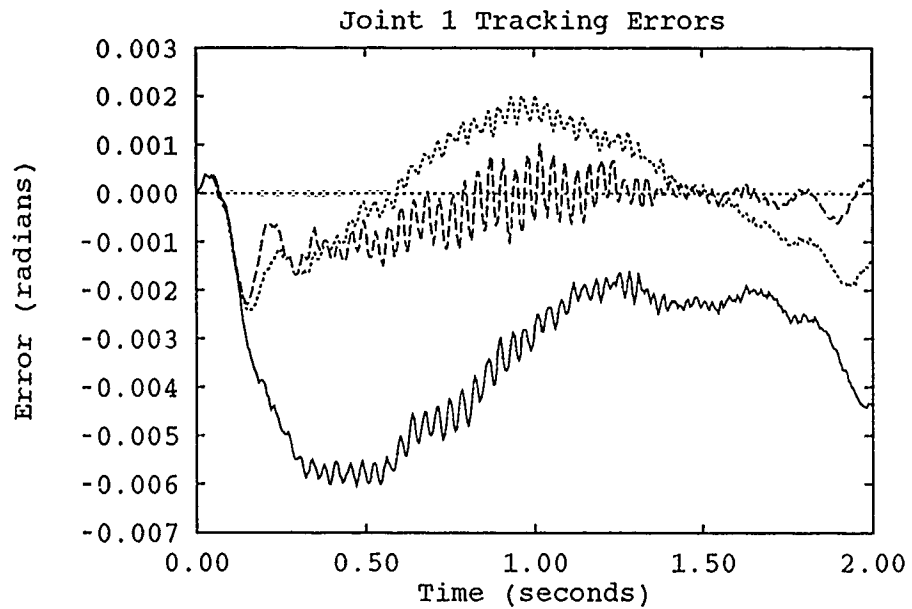


Figure R.16. Comparison of MBAIC and SMBC Controllers - Trajectory 4

-----	Tarokh-Based MBAIC	————	SMBC Controller
.....	Seraji-Based MBAIC		

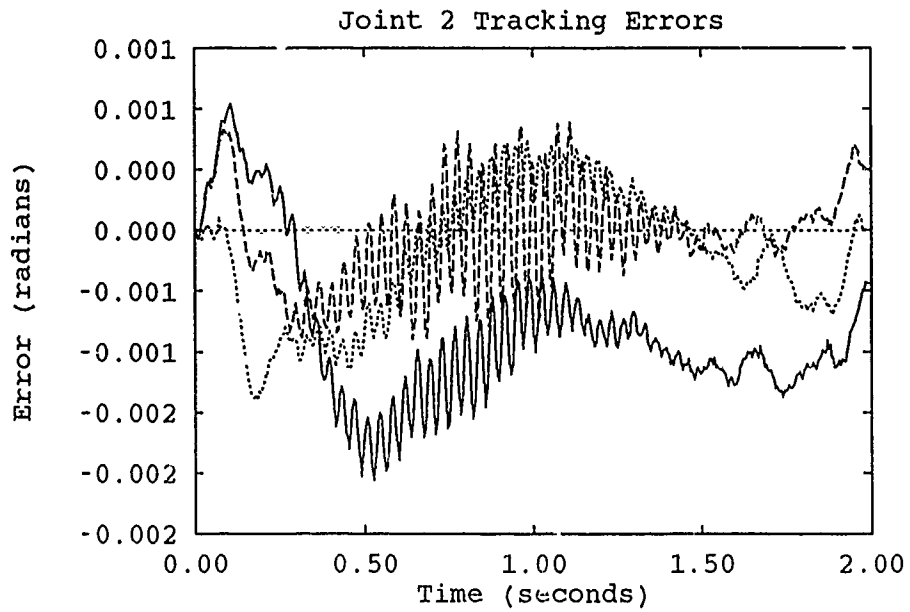


Figure R.17. Comparison of MBAIC and SMBC Controllers - Trajectory 4

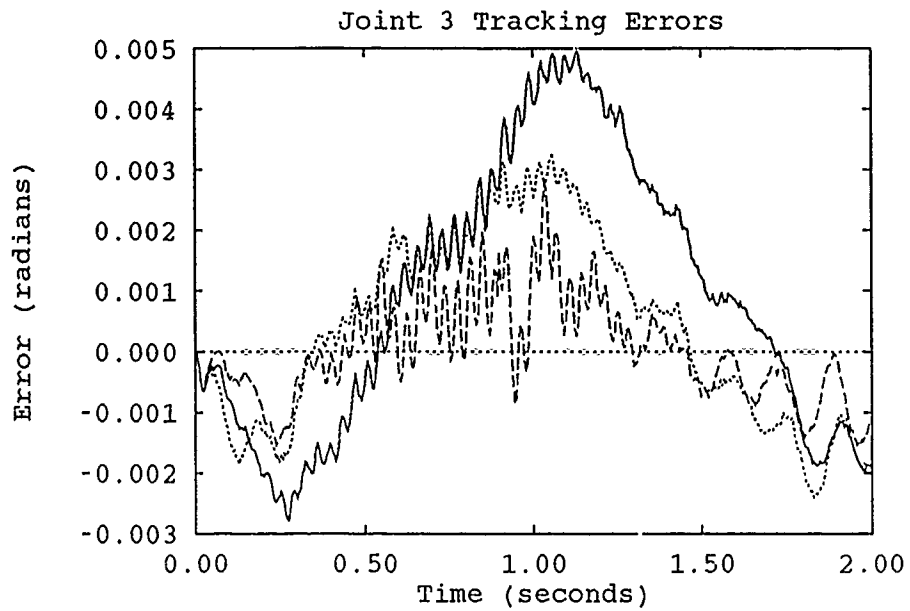


Figure R.18. Comparison of MBAIC and SMBC Controllers - Trajectory 4

-----	Tarokh-Based MBAIC	————	SMBC Controller
.....	Seraji-Based MBAIC		

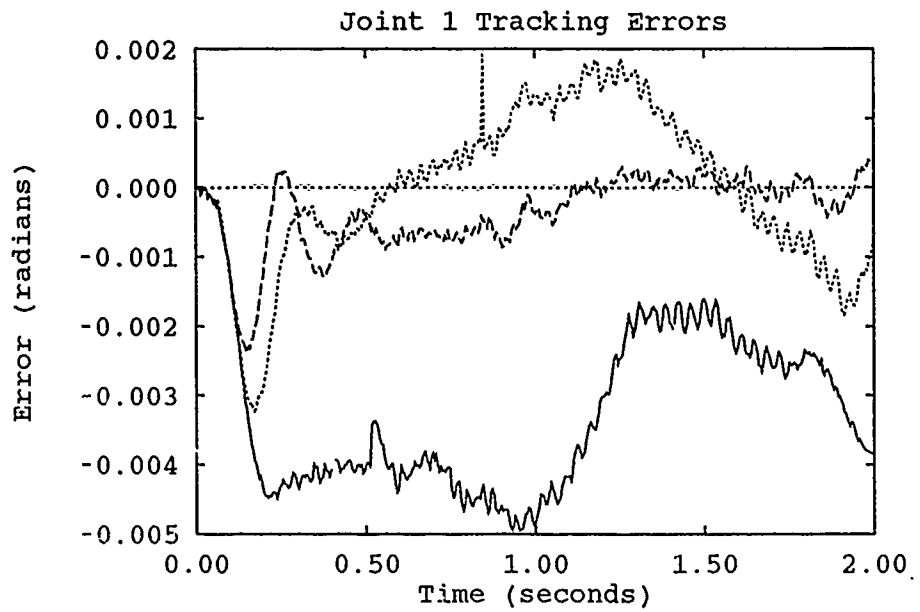


Figure R.19. Comparison of MBAIC and SMBC Controllers - Trajectory 5

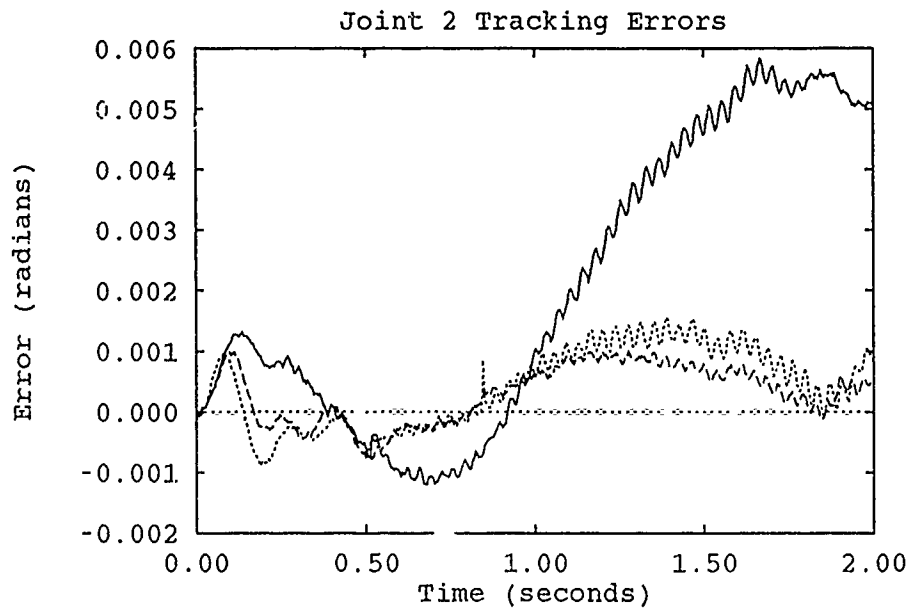


Figure R.20. Comparison of MBAIC and SMBC Controllers - Trajectory 5

-----	Tarokh-Based MBAIC	————	SMBC Controller
.....	Seraji-Based MBAIC		

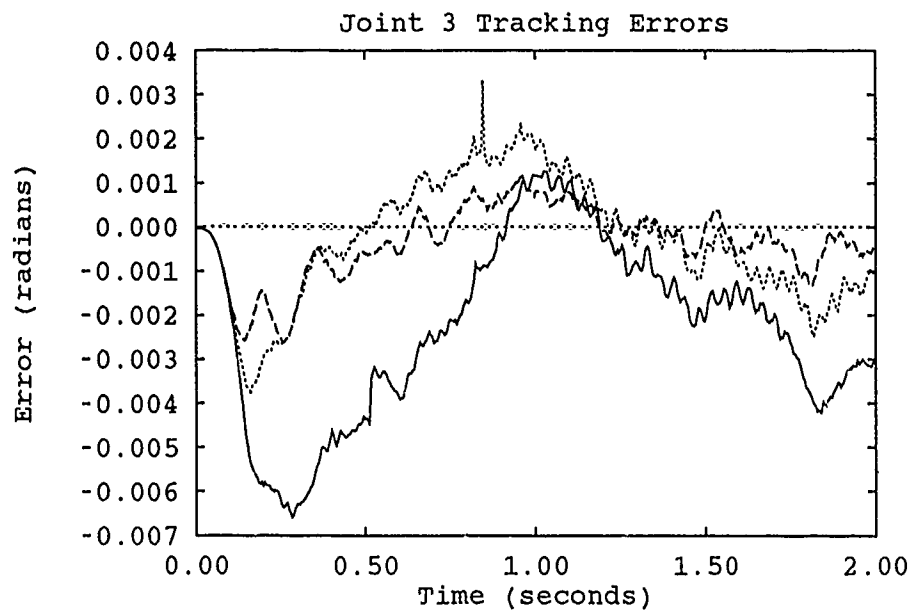


Figure R.21. Comparison of MBAIC and SMBC Controllers - Trajectory 5

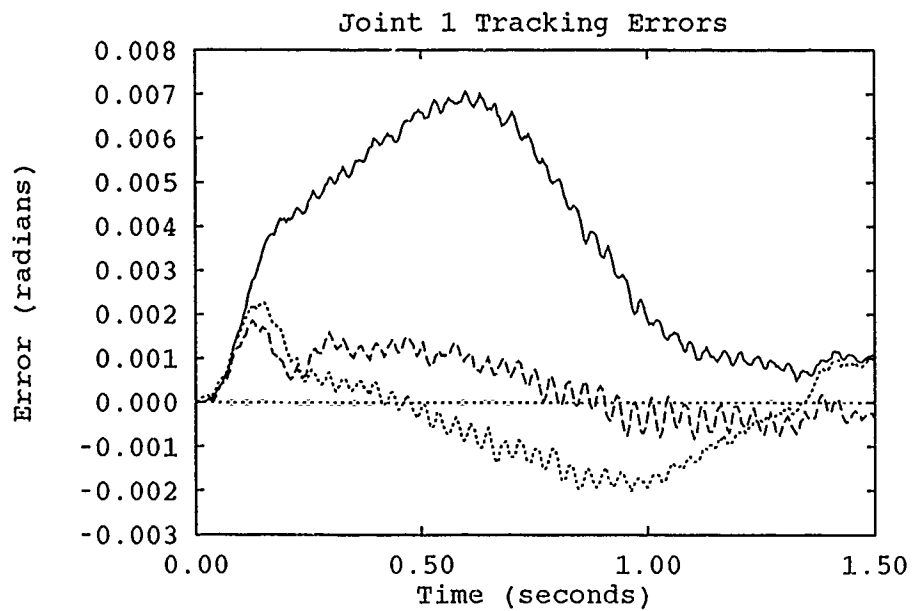


Figure R.22. Comparison of MBAIC and SMBC Controllers - Trajectory 1

----	Tarokh-Based MBAIC	—	SMBC Controller
.....	Seraji-Based MBAIC		

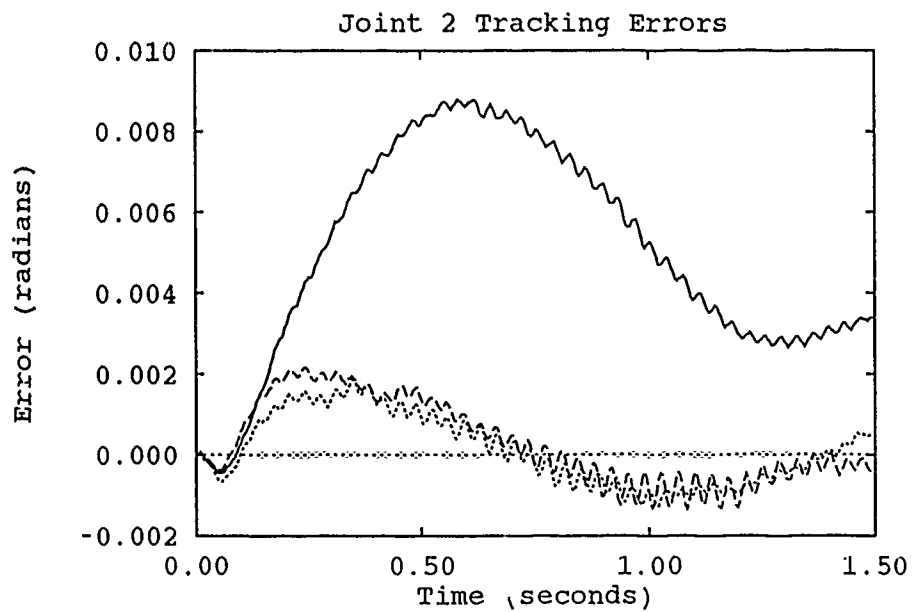


Figure R.23. Comparison of MBAIC and SMBC Controllers - Trajectory 1

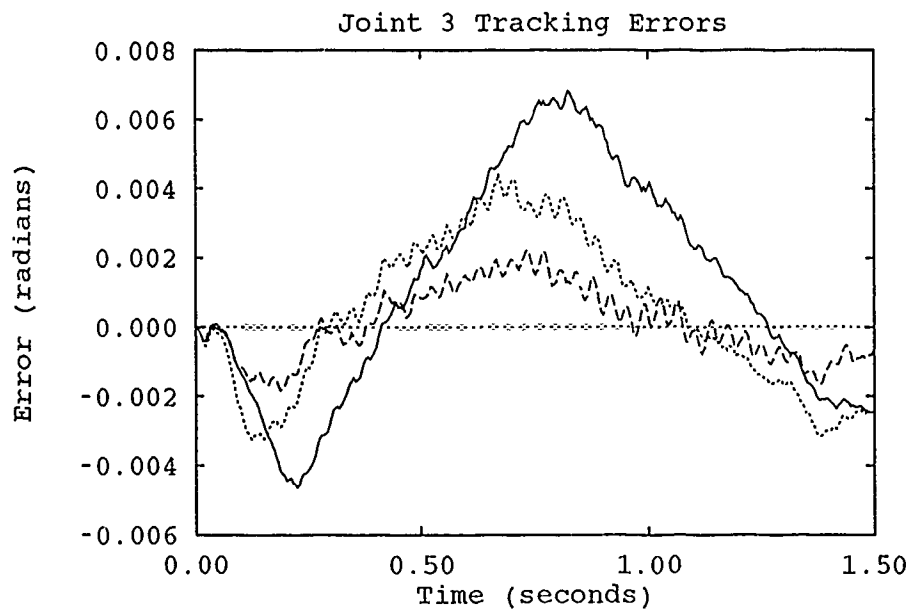


Figure R.24. Comparison of MBAIC and SMBC Controllers Trajectory 1

---	Tarokh-Based MBAIC	—	SMBC Controller
.....	Seraji-Based MBAIC		

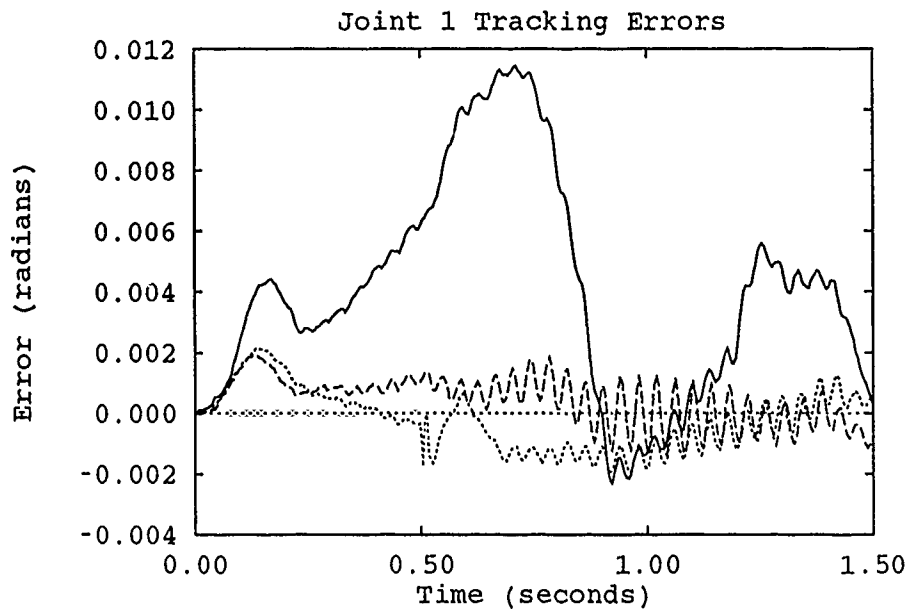


Figure R.25. Comparison of MBAIC and SMBC Controllers - Trajectory 1 w/ Payload

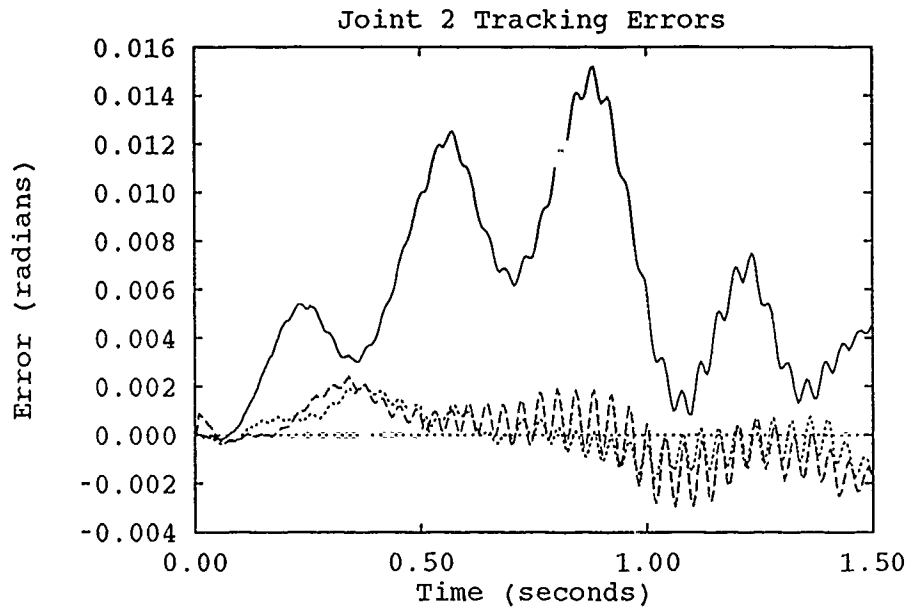


Figure R.26. Comparison of MBAIC and SMBC Controllers Trajectory 1 w/ Payload

----	Tarokh-Based MBAIC	—	SMBC Controller
.....	Seraji-Based MBAIC		

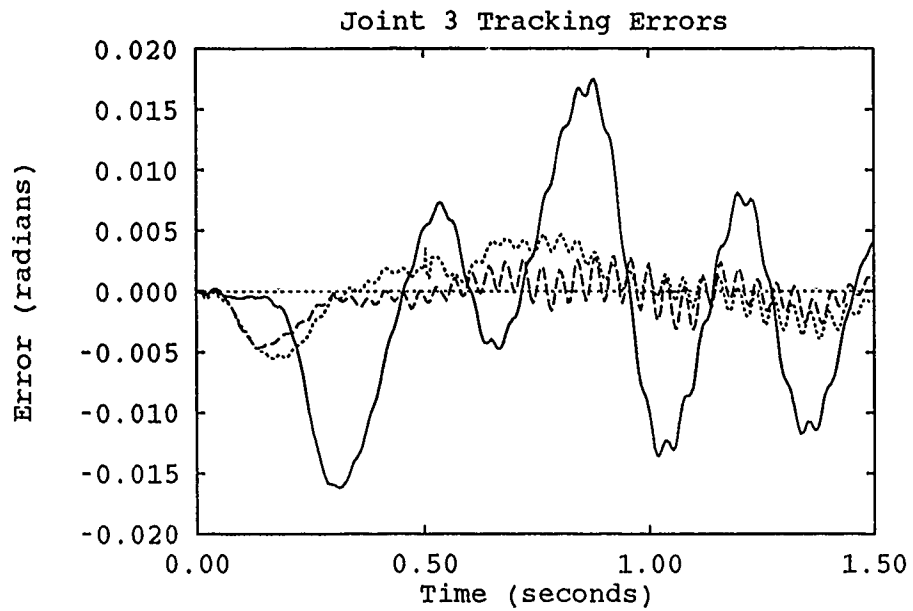


Figure R.27. Comparison of MBAIC and SMBC Controllers - Trajectory 1 w/ Payload

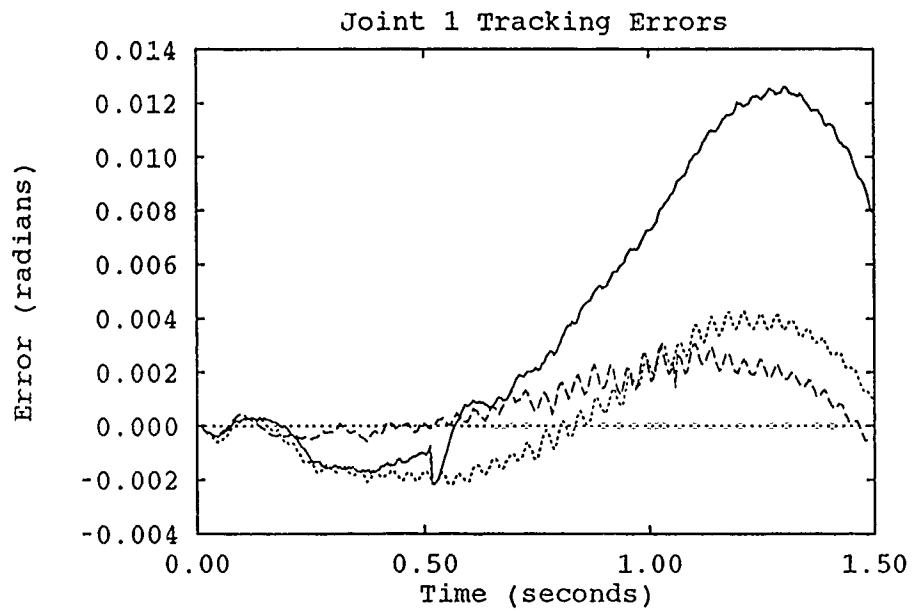


Figure R.28. Comparison of MBAIC and SMBC Controllers Trajectory 6

-----	Tarokh-Based MBAIC	————	SMBC Controller
.....	Seraji-Based MBAIC		

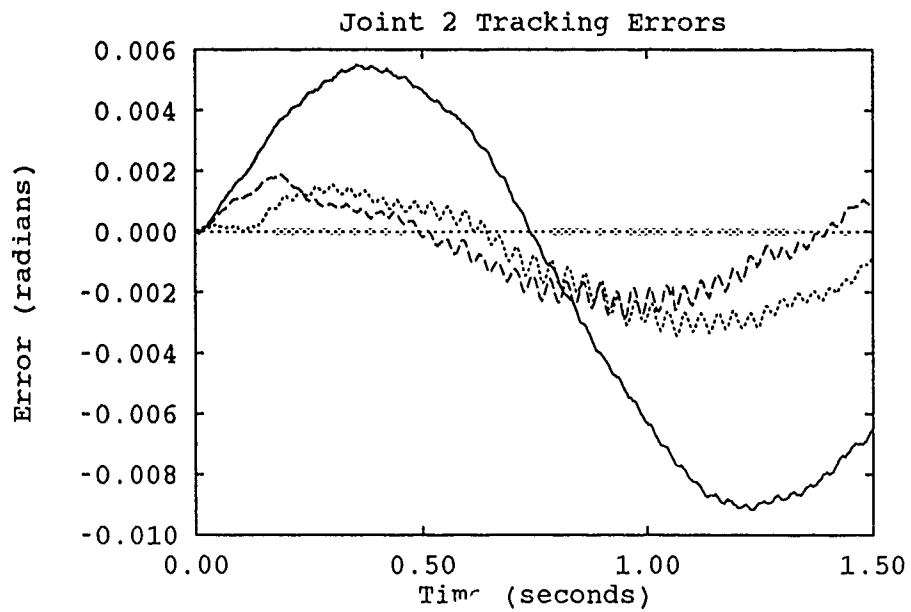


Figure R.29. Comparison of MBAIC and SMBC Controllers - Trajectory 6

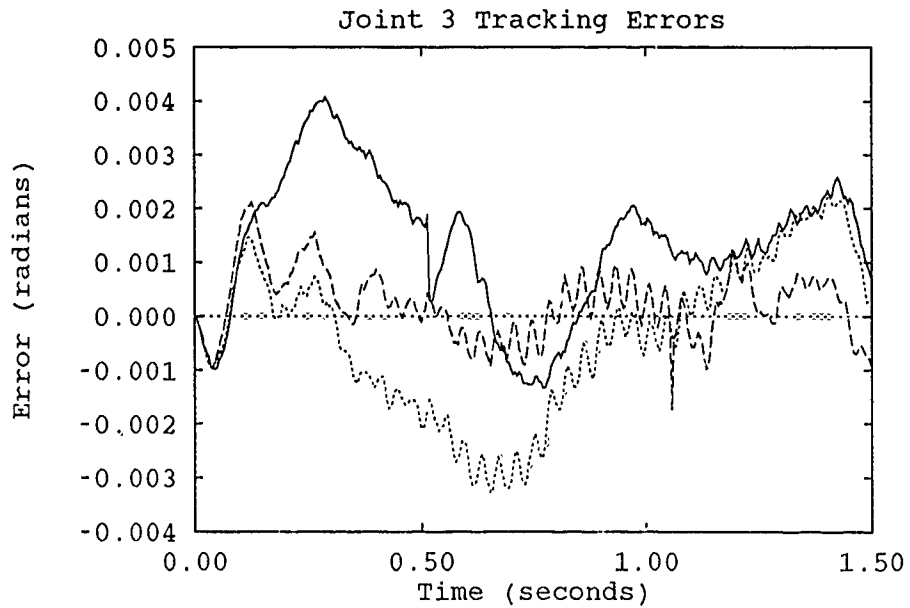


Figure R.30. Comparison of MBAIC and SMBC Controllers - Trajectory 6

-----	Tarokh-Based MBAIC	————	SMBC Controller
.....	Seraji-Based MBAIC		

Appendix S. *Comparison of MBAIC and AMBC Controllers*

This appendix contains plots which compare the results achieved in 19-parameter AMBC algorithm testing with the trajectory errors produced in MBAIC testing.

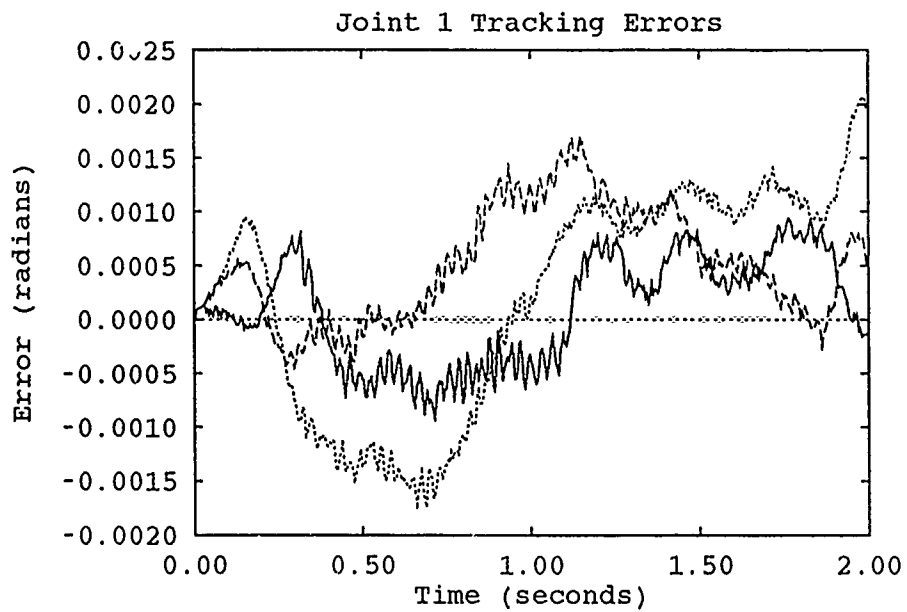


Figure S.1. Comparison of MBAIC and AMBC Controllers - Trajectory 2

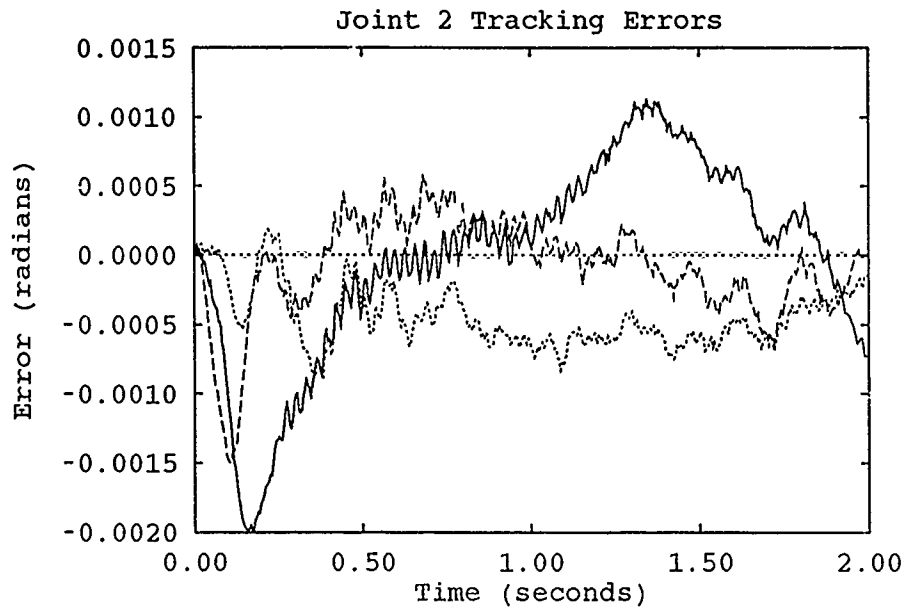


Figure S.2. Comparison of MBAIC and AMBC Controllers Trajectory 2

-----	Tarokh-Based MBAIC	———	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

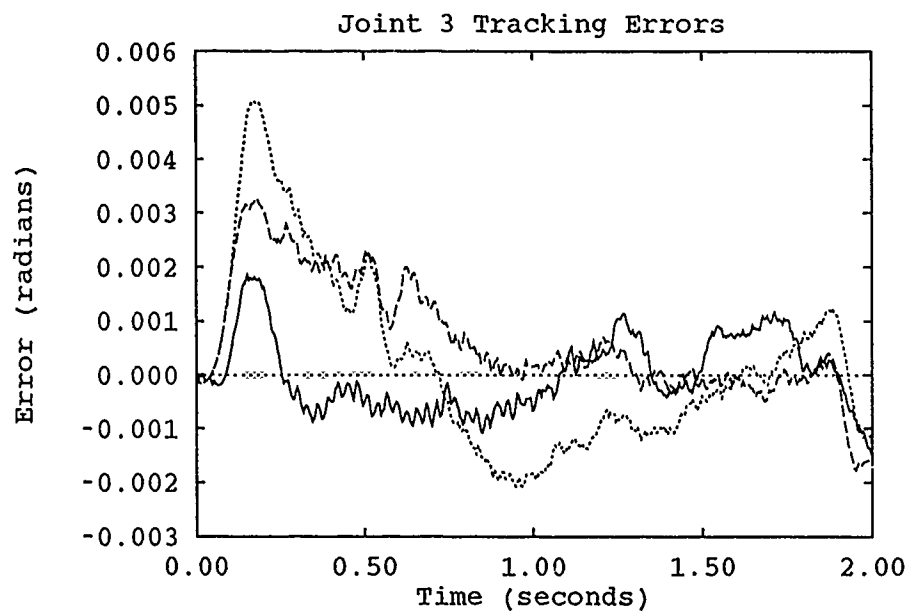


Figure S.3. Comparison of MBAIC and AMBC Controllers - Trajectory 2

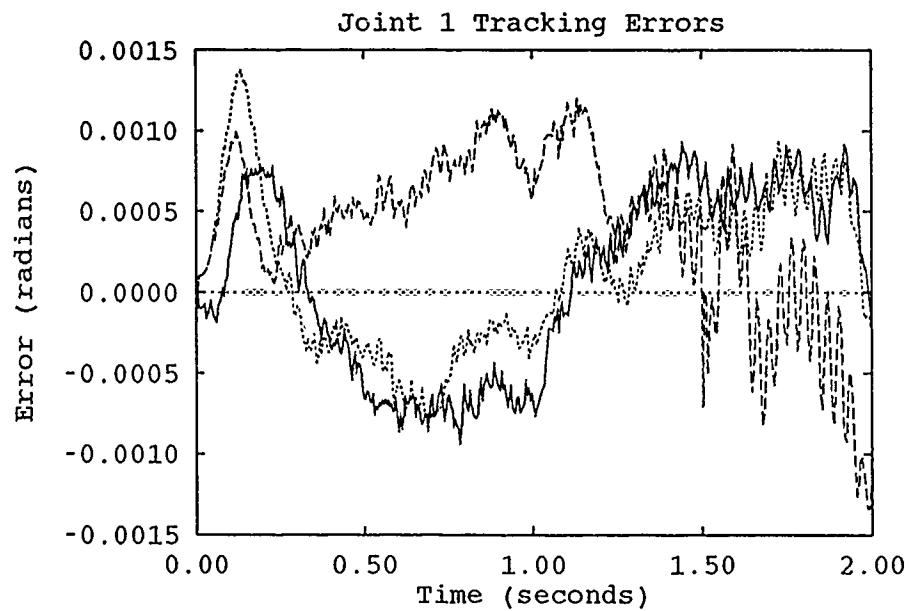


Figure S.4. Comparison of MBAIC and AMBC Controllers - Trajectory 3

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

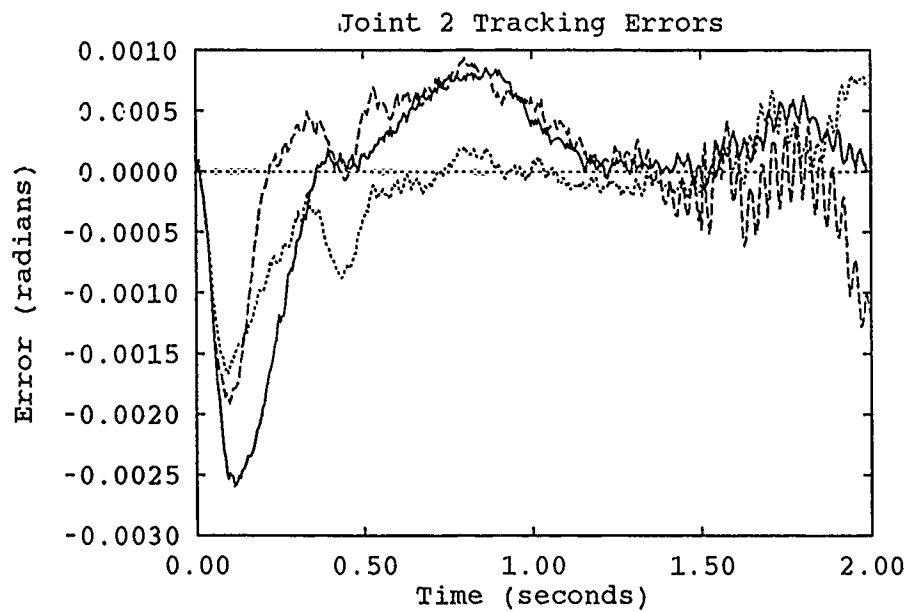


Figure S.5. Comparison of MBAIC and AMBC Controllers - Trajectory 3

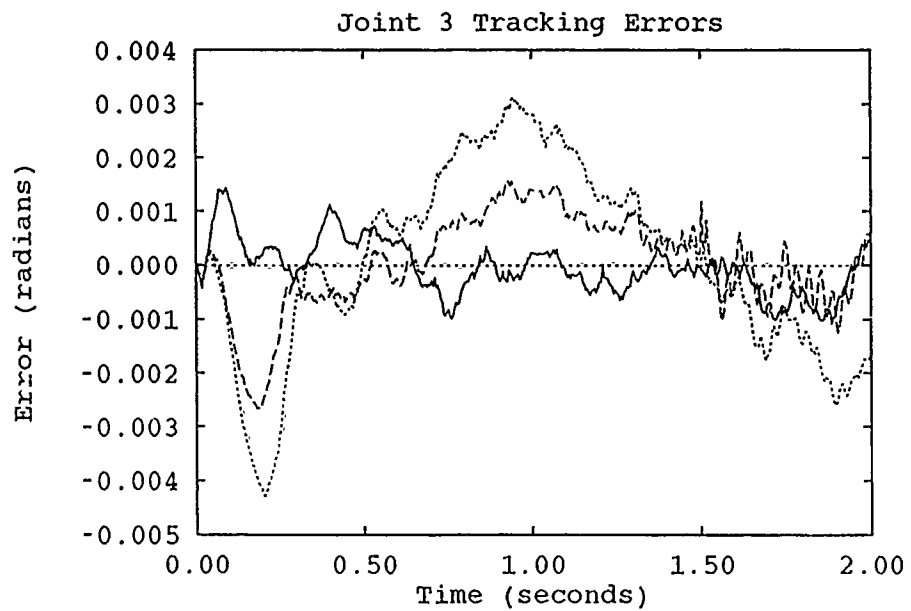


Figure S.6. Comparison of MBAIC and AMBC Controllers - Trajectory 3

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

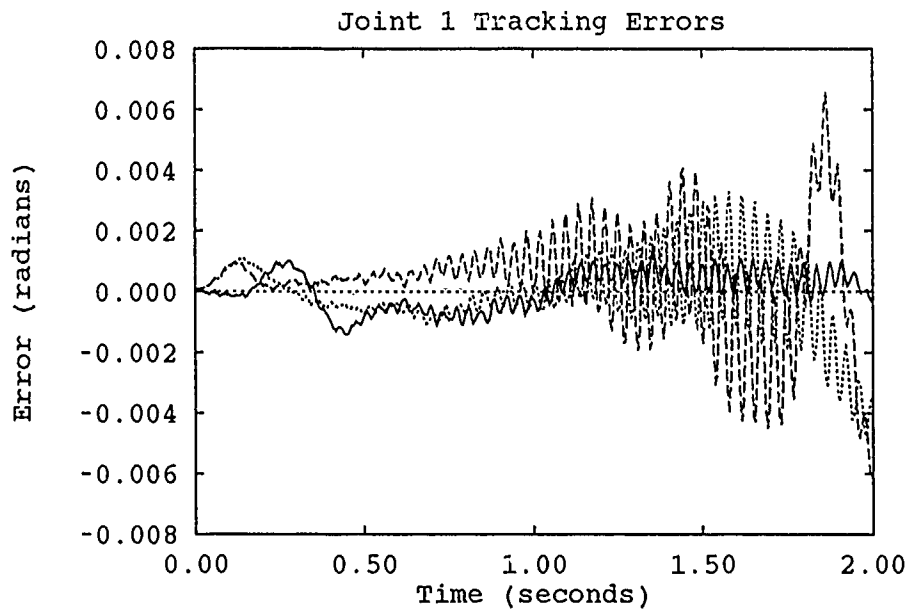


Figure S.7. Comparison of MBAIC and AMBC Controllers - Trajectory 3 w/ Payload

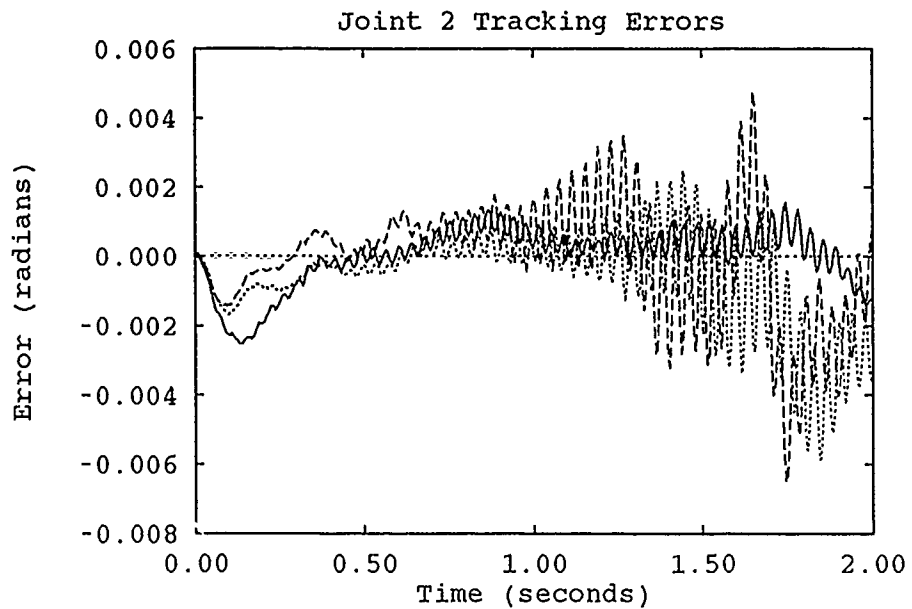


Figure S.8. Comparison of MBAIC and AMBC Controllers Trajectory 3 w/ Payload

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

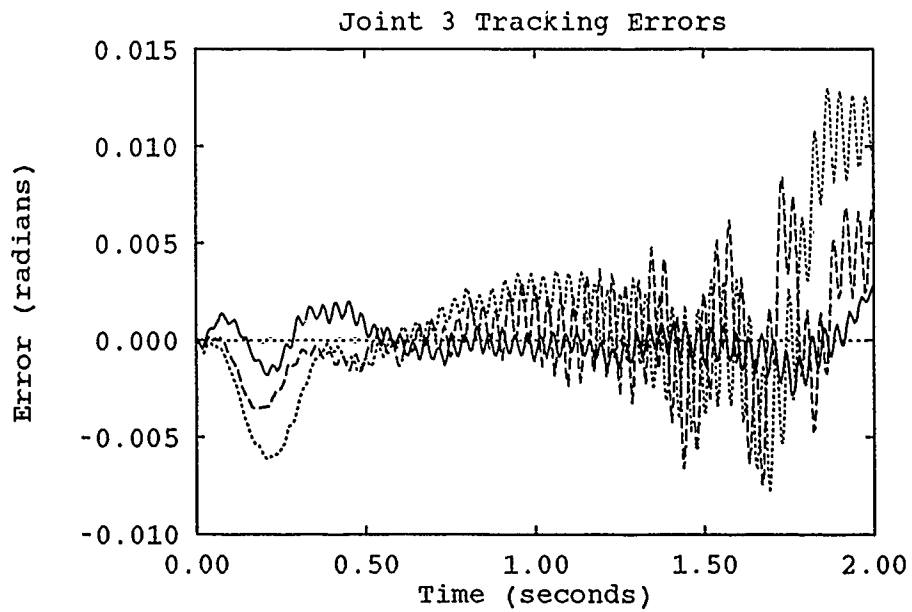


Figure S.9. Comparison of MBAIC and AMBC Controllers - Trajectory 3 w/ Payload

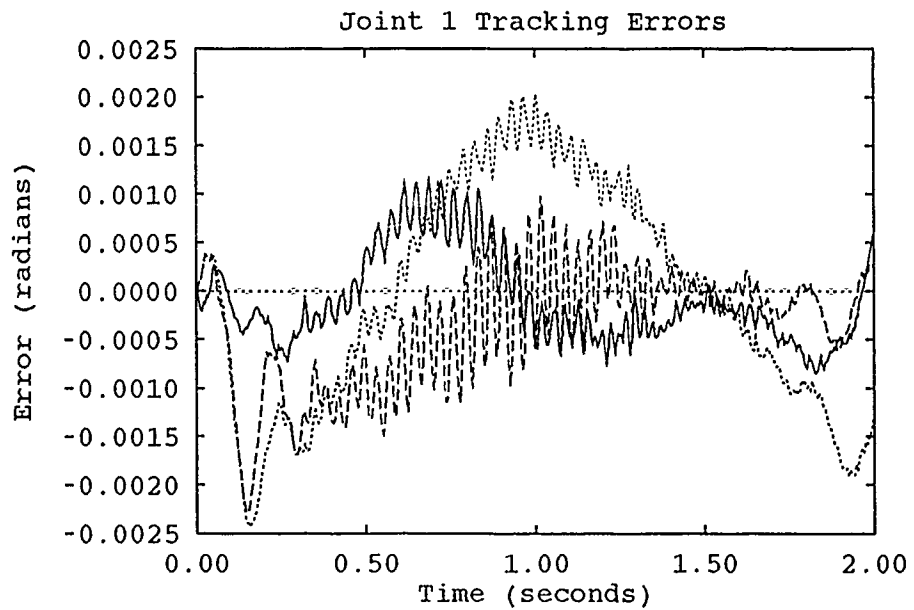


Figure S.10. Comparison of MBAIC and AMBC Controllers - Trajectory 4

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

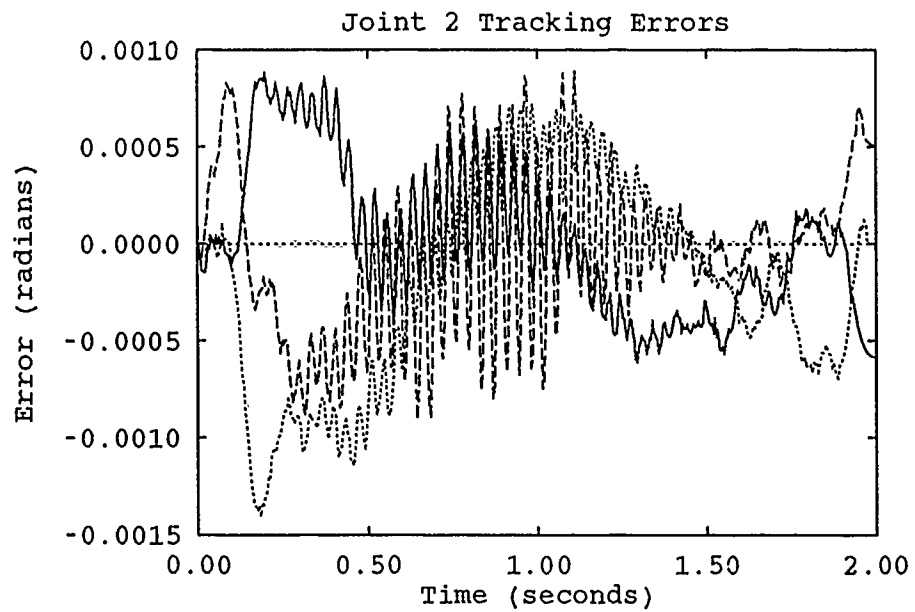


Figure S.11. Comparison of MBAIC and AMBC Controllers - Trajectory 4

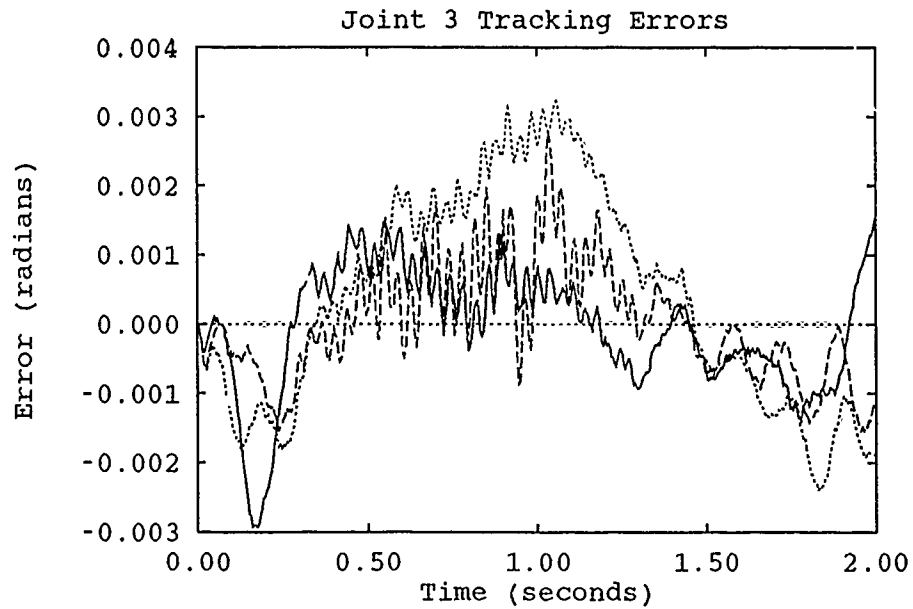


Figure S.12. Comparison of MBAIC and AMBC Controllers - Trajectory 4

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

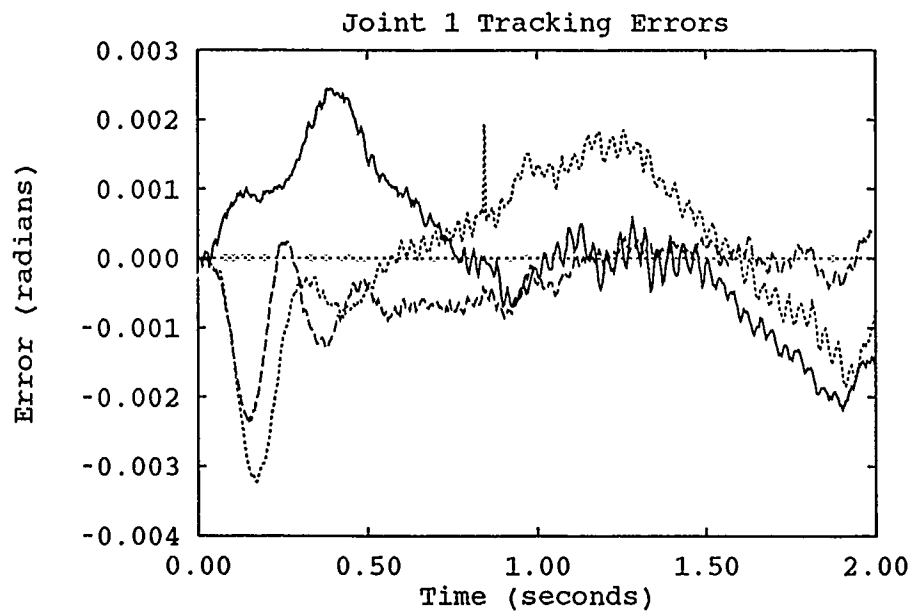


Figure S.13. Comparison of MBAIC and AMBC Controllers - Trajectory 5

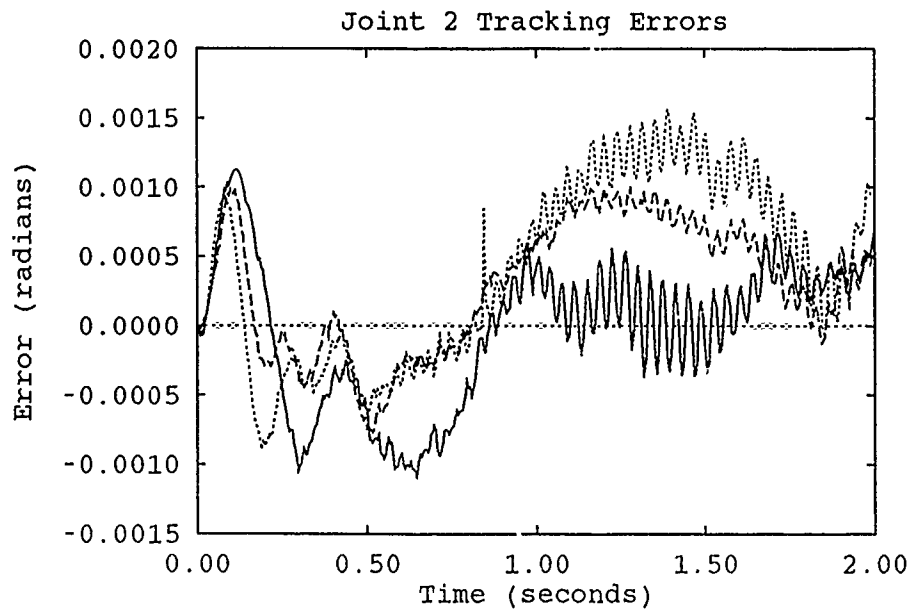


Figure S.14. Comparison of MBAIC and AMBC Controllers - Trajectory 5

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

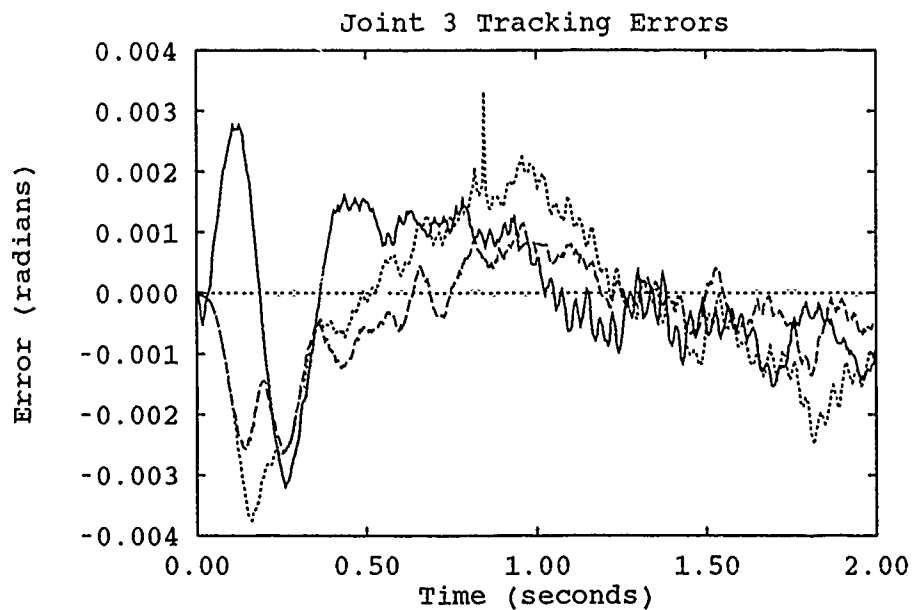


Figure S.15. Comparison of MBAIC and AMBC Controllers - Trajectory 5

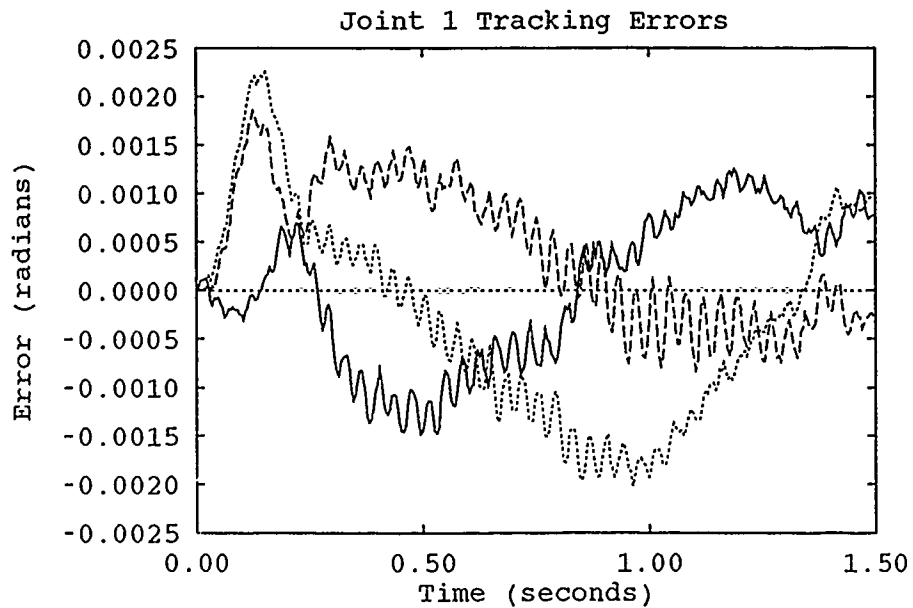


Figure S.16. Comparison of MBAIC and AMBC Controllers - Trajectory 1

-----	Tarokh-Based MBAIC	————	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

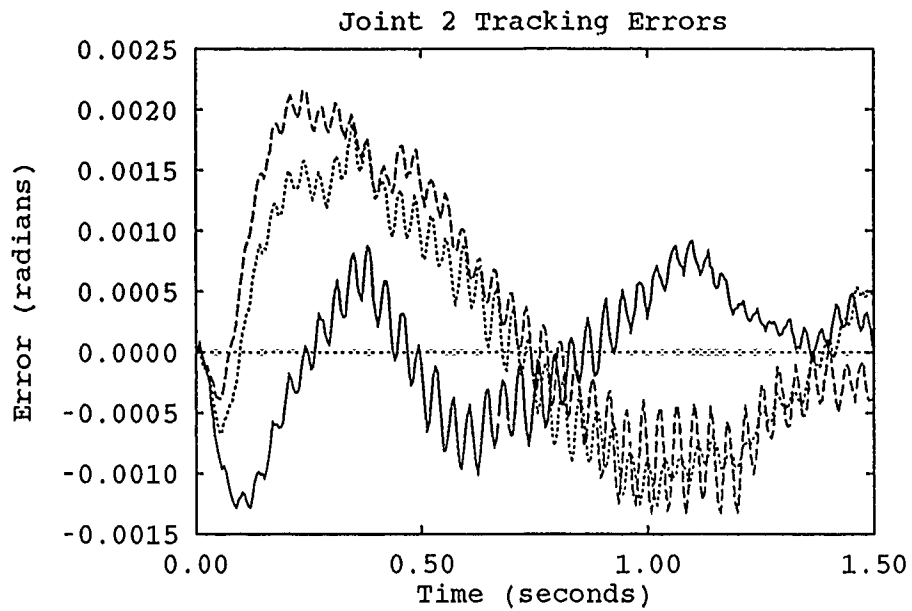


Figure S.17. Comparison of MBAIC and AMBC Controllers - Trajectory 1

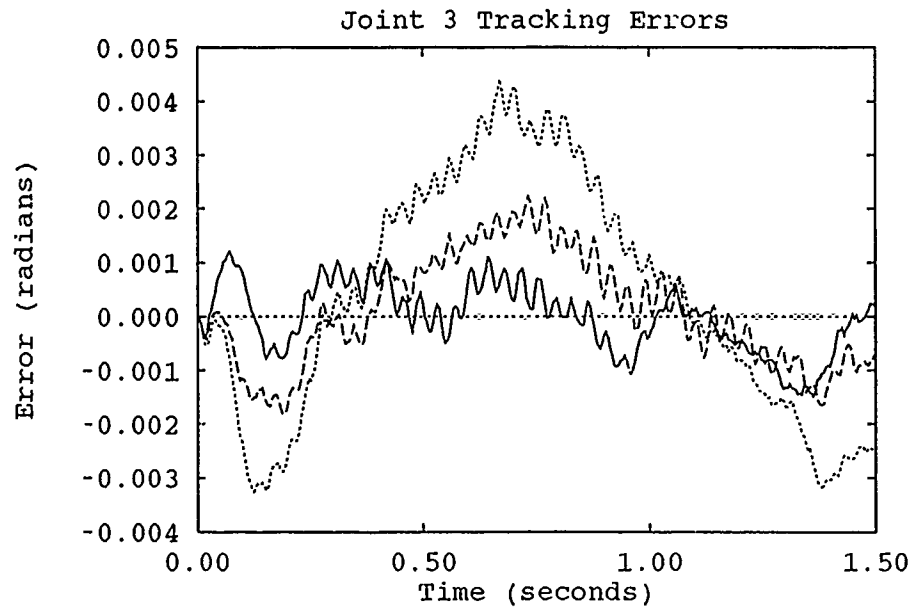


Figure S.18. Comparison of MBAIC and AMBC Controllers Trajectory 1

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

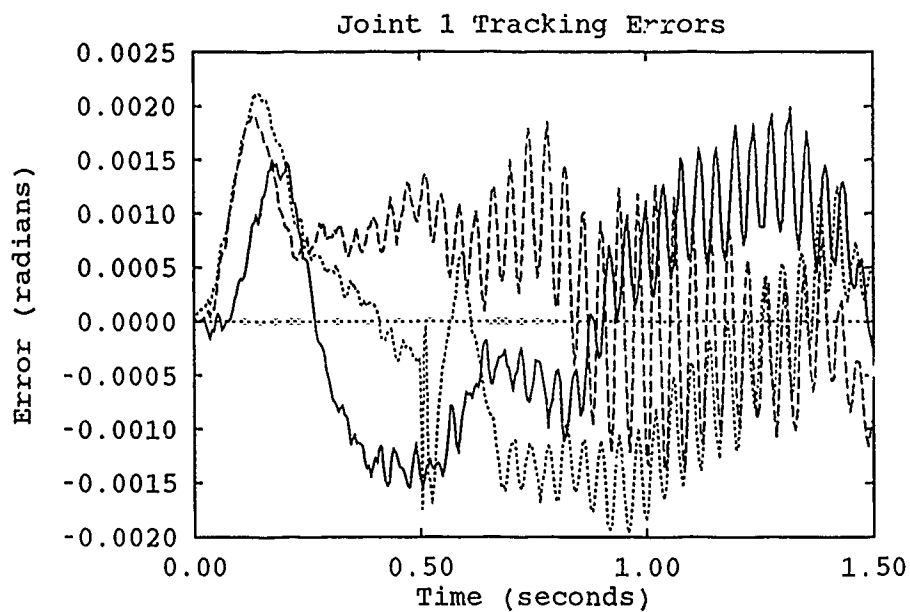


Figure S.19. Comparison of MBAIC and AMBC Controllers - Trajectory 1 w/ Payload

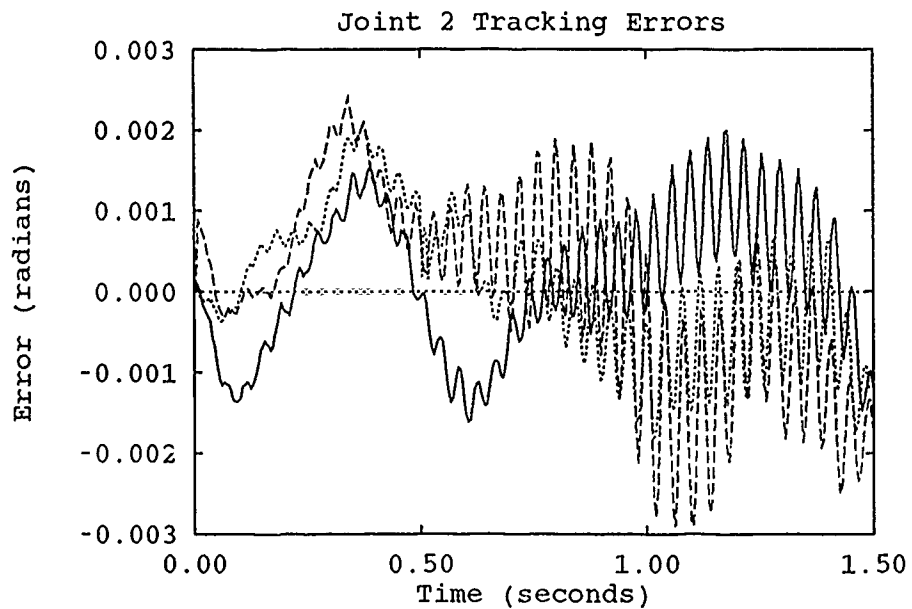


Figure S.20. Comparison of MBAIC and AMBC Controllers - Trajectory 1 w/ Payload

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

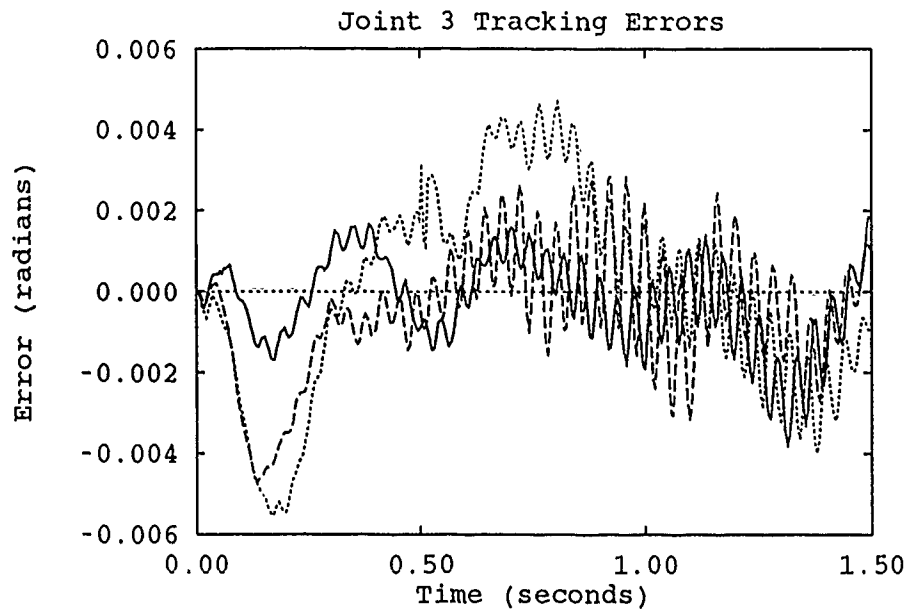


Figure S.21. Comparison of MBAIC and AMBC Controllers - Trajectory 1 w/ Payload

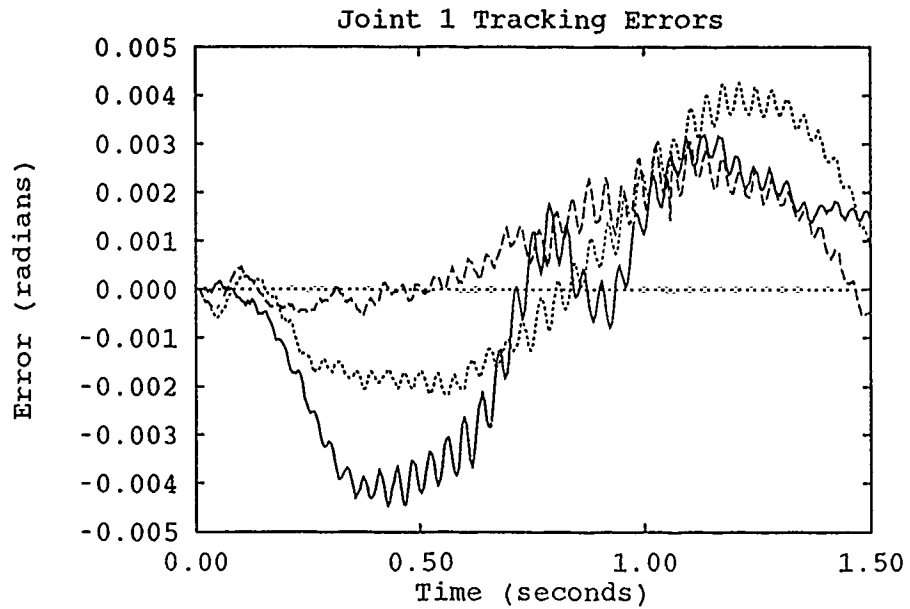


Figure S.22. Comparison of MBAIC and AMBC Controllers - Trajectory 6

----	Tarokh-Based MBAIC	—	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		

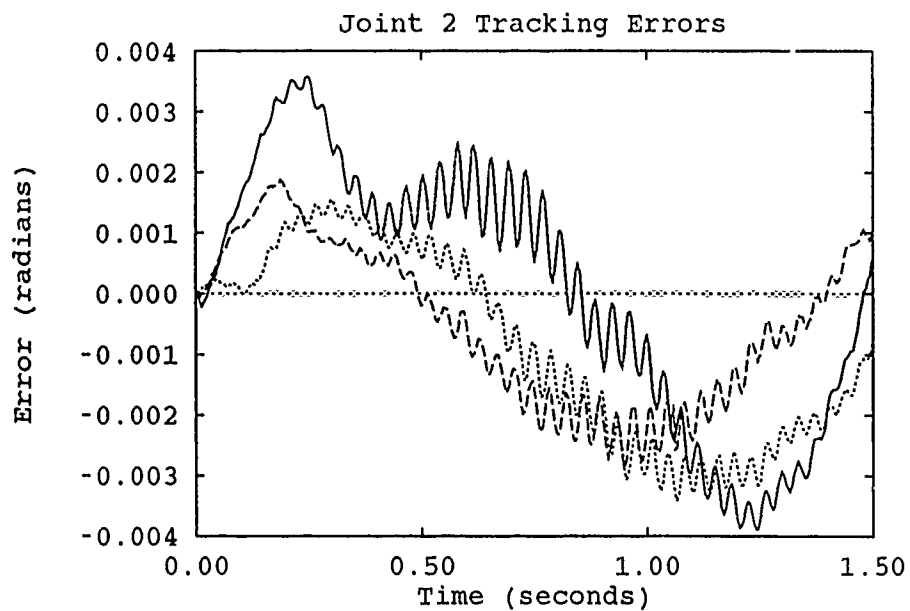


Figure S.23. Comparison of MBAIC and AMBC Controllers - Trajectory 6

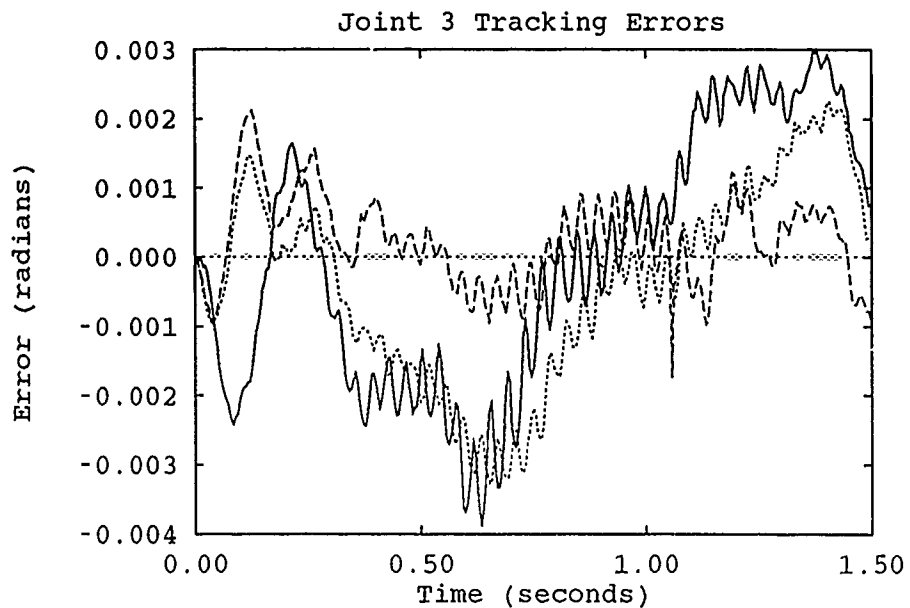


Figure S.24. Comparison of MBAIC and AMBC Controllers - Trajectory 6

-----	Tarokh-Based MBAIC	————	19-Parameter AMBC Controller
.....	Seraji-Based MBAIC		